Henry C. Bunsow (SBN 60707)
K.T. Cherian (SBN 133967)
HOWREY LLP
525 Market Street, Suite 3600
San Francisco, CA 94105
Telephone: (415) 848-4900
Facsimile: (415) 848-4999
E-Mail: BunsowH@howrey.com
E-Mail: CherianK@howrey.com

James C. Pistorino (SBN 226496)
James F. Valentine (SBN 149269)
HOWREY LLP
1950 University Avenue, 4th Floor
East Palo Alto, CA 94303
Telephone: (650) 798-3500
Facsimile: (650) 798-3600
E-Mail: PistorinoJ@howrey.com
E-Mail: ValentineJ@howrey.com

Attorneys for Plaintiffs
SEVEN NETWORKS INTERNATIONAL OY



UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

EMC

| | |
|---|---|
| SEVEN NETWORKS INTERNATIONAL OY (formerly Smartner Information Systems, Ltd.), a Finnish corporation,<br><br>Plaintiff,<br><br>vs.<br><br>VISTO CORPORATION, a Delaware corporation,<br><br>Defendant. | Case No. PVT 07 0783<br><br>**COMPLAINT FOR DECLARATORY JUDGMENT** |

Plaintiff Seven Networks International OY ("SNIO") (formerly Smartner Information Systems, Ltd.), for its complaint against Defendant Visto Corporation ("Visto"), alleges and avers:

## PARTIES

1. Seven Networks International OY (SNIO) is a Finnish corporation having its principal place of business in Helsinki, Finland. Until April 2005, SNIO was known as Smartner Information Systems, Ltd. In order to avoid confusion with another company, SNIO will be referred to herein as "Smartner/SNIO."

2. Visto is a Delaware corporation having its principal place of business at 275 Shoreline Drive, Suite 300, Redwood Shores, California 94065.

## JURISDICTION AND VENUE

3. This is an action for the resolution of an existing conflict under the Declaratory Judgment Act, 28 U.S.C. §§ 2201 and 2202. The underlying causes of action arise under the patent laws of the United States. A case or controversy exists between Plaintiff and Visto. The amount in controversy between the parties exceeds $75,000. This Court therefore has subject matter jurisdiction under 28 U.S.C. §§ 1331, 1332, and 1338(a).

4. On information and belief, this Court has personal jurisdiction over Visto because Visto is found in this District.

5. Venue for this action is proper in this District under 38 U.S.C. §§ 1391(b) and 1400(b) because Visto resides in this District and because a substantial part of the events giving rise to this claim occurred in this District.

## PRIOR LITIGATION BETWEEN SMARTNER/SNIO AND VISTO

6. Pending in the Eastern District of Texas are two related cases captioned *Visto Corporation v. Smartner Information Systems, Ltd.*, Civil Action No. 2:05-CV-91-TJW, filed on February 25, 2005 (hereinafter, "the *Smartner* case" or "*Smartner*") and *Seven Networks, Inc. v. Visto Corporation*, Civil Action No. 2:05-CV-365-TJW, filed on August 10, 2005 ("the *Seven* case" or "*Seven*"). In the *Smartner* case, Visto is accusing Smartner/SNIO of infringing three United States Patents. In the *Seven* case, Seven is accusing Visto of infringing two United States Patents owned by Seven.

7. In June 2006, Visto threatened to amend its complaint in the *Smartner* case and its answer in the *Seven* case to allege infringement of Visto's U.S. Patent Nos. 6,151,606 (the "'606 patent") and 7,039,679 (the "'679 patent") by Smartner/SNIO and Seven. (*See* Exhibits A and B hereto.) In anticipation of Visto's motion for leave to amend, Smartner/SNIO and Seven filed a declaratory judgment action in this Court seeking a declaration that the '606 and '679 patents were not infringed by either company and were unenforceable and invalid. (*See Seven Networks, Inc. v. Visto Corporation*, N.D. Cal. Case No. 3:06-CV-03650-WHA, Docket Entry No. 1.) Subsequently, Visto did move for leave to amend in both of the Eastern District of Texas cases, which both Smartner/SNIO and Seven opposed.

8. Visto then filed a motion to dismiss or transfer the case pending in this Court in favor of the pending cases in the Eastern District of Texas. (*See Seven v. Visto*, 3:06-CV-03650-WHA, Docket Entry Nos. 10-12.)

9. On August 17, 2006, Judge Ward granted Visto's motion for leave to amend in the *Seven* case on the grounds that Visto was the first to file because it sought a meet and confer prior to the time that Seven filed its declaratory judgment action in this Court. (*See* Exhibit C.)

10. In light of Judge Ward's reasoning and the potential for a conflict among the Districts on the "first to file rule," Seven and Smartner/SNIO informed this Court that they did not oppose Visto's motion to dismiss or transfer. Accordingly, on August 29, 2006, this Court dismissed the actions. (*See Seven v. Visto*, 3:06-CV-03650-WHA, Docket Entry No. 14.)

11. However, on January 26, 2007, Visto served notice that it was withdrawing its motion for leave to assert the '606 and '679 patents in the *Smartner* case mentioning that the court had already completed its *Markman* proceedings on the patents originally asserted. On January 31, 2007, in light of Visto's notice, Judge Ward denied Visto's motion for leave to amend to assert the '606 and '679 patents against Smartner/SNIO. (*See* Exhibit D.) Accordingly, there is no presently filed case where Visto's allegation of infringement by Smartner/SNIO is at issue, and Smartner/SNIO seeks a determination of that issue.

**SMARTNER/SNIO'S REASONABLE APPREHENSION OF SUIT**

12. This action is brought to resolve the apprehension under which Smartner/SNIO is forced to conduct its business as a result of Visto's threats to sue Smartner/SNIO for infringement of certain patents (the '606 and '679 patents) purportedly owned by Visto.

13. Smartner/SNIO is a leading designer, manufacturer, and marketer of innovative wireless solutions for the worldwide mobile communications market. Smartner/SNIO's portfolio of award-winning products is used by numerous organizations around the world and includes the Always-on-Mail and Duality wireless platforms, software development tools, and software/hardware licensing agreements.

14. As detailed above, Visto has actually accused Smartner/SNIO of infringing the '606 and '679 patents. Visto's allegations of infringement of the '606 and '679 patents have created in Smartner/SNIO a reasonable apprehension that Visto will again sue Smartner/SNIO for patent infringement of the '606 and '679 Visto patents. Smartner/SNIO believes that failure to determine the issues presented by this case at this point in time will lead to substantial commercial injury to Smartner/SNIO.

15. Smartner/SNIO therefore seeks a declaration by this Court that Smartner/SNIO's products and services do not infringe the '606 and '679 Visto patents, and that the '606 and '679 Visto patents are invalid and unenforceable.

**COUNT I**

**Declaratory Judgment of Noninfringement of the '606 Patent**

16. Smartner/SNIO repeats and realleges paragraphs 1 through 15 of this Complaint as if the same were full set forth herein.

17. Smartner/SNIO's products (including its Duality and Always-on-Mail products) do not infringe any valid claim of the '606 patent, either directly, indirectly, contributorily, or otherwise. Seven has not induced others to infringe the '606 patent.

18. Smartner/SNIO is therefore entitled to a declaratory judgment that it does not infringe the '606 patent.

**COUNT II**

**Declaratory Judgment of Invalidity of the '606 Patent**

19. Smartner/SNIO repeats and realleges paragraphs 1 through 18 of this Complaint as if the same were full set forth herein.

20. The claims of the '606 patent are invalid for failure to meet the requirements specified in Title 35 of the United States Code, including, but not limited to, 35 U.S.C. §§ 101, 102, 103, and 112.

21. Smartner/SNIO is therefore entitled to a declaratory judgment that the '606 patent is invalid.

**COUNT III**

**Declaratory Judgment of Unenforceability of the '606 Patent**

22. Smartner/SNIO repeats and realleges paragraphs 1 through 21 of this Complaint as if the same were full set forth herein.

23. The claims of the '606 patent are unenforceable by reason of their having been procured through inequitable conduct and fraud. Particularly, the applicants failed to advise the Examiner for the '606 patent that U.S. Patent Nos. 6,023,708 (the "'708 patent"), 5,961,590 (the "'590 patent"), 5,968,131 (the "'131 patent"), and 6,131,116 (the "'116 patent"), patents of which the '679 claims priority, had been rejected, and the bases for the prior rejection by the Examiners for those patents. As such, the applicants knew of and were guilty of intentionally concealing material information from the USPTO concerning the prosecution histories of the '708, '590, '131, and '116 patents. The rejected claims of the '708, '590, '131, and '116 applications and the claims of the '679 application are similar claims in a similar technology. On October 15, 1998, the USPTO rejected all of the claims in the '708 application in light of various pieces of prior art. Those pieces of prior art included U.S. Patent No. 5,790,790 to Smith et al. ("Smith"), U.S. Patent No. 5,721,908 to Lagarde et al. ("Lagarde"), U.S. Patent No. 5,799,318 to Cardinal et al. ("Cardinal"), and U.S. Patent No. 5,875,159 to Cary et al. ("Cary"). On February 24, 1998. the USPTO rejected all the claims in the '590 application in light of U.S. Patent No. 5,647,022 to Brunson ("Brunson"). On April 14, 1998, the USPTO rejected all of the

claims in the '116 application in light of various pieces of prior art. Those pieces of prior art included U.S. Patent No. 5,706,502 to Foley et al. ("Foley") and Using Netscape 2. The USPTO further rejected all of the claims in the '116 application on October 27, 1998, in light of U.S. Patent No. 5,812,668 to Weber ("Weber") and U.S. Patent No. 5,768,510 to Gish ("Gish"), and on April 12, 1999, in light of U.S. Patent No. 5,828,840 to Cowan et al. ("Cowan"). On January 25, 1999, the USPTO rejected claims 28-46 of the '131 application in light of U.S. Patent No. 5,758,355 to Buchanan ("Buchanan"). On October 26, 1998, the attorney of record for the patentee submitted an Information Disclosure Statement ("IDS") for the '606 patent to the USPTO. The IDS listed the Smith, Lagarde, Cardinal, and Cary art. On October 27, 1998, a supplemental IDS for the '606 patent was submitted to the USPTO, listing the Foley art. The Brunson and Buchanan art was disclosed to the USPTO via supplemental IDS's on January 11, 1999, and February 16, 1999, respectively. However, the attorney of record failed to apprise the Examiner that the cited art was used in the rejection of the '708, '116, '131, and '590 applications. Therefore, Visto had knowledge of information material to the patentability of the '606 application and failed to disclose that information to the USPTO.

24. Furthermore, during the prosecution of the '606 patent, the applicants failed to disclose to the Examiner the Lotus Notes software program, literature related to Lotus Notes, and the Intellink/IntelliSync software programs and literature. The inventors (*e.g.*, David Cowan) and those associated with the prosecution of the applications of the patents-in-suit were aware of the Lotus Notes prior art and failed to disclose it to the Examiner. Likewise, the inventors (*e.g.*, Daniel Méndez) and those associated with the prosecution of the applications of the patents-in-suit were aware of prior art synchronizing translators and failed to disclose them to the Examiner. U.S. Patent Nos. 5,812,668, 5,768,510, 5,828,840, and Using Netscape 2, references cited by Examiners in co-pending applications, were likewise not disclosed to the Examiner of the '606 patent. Each of these references (and the other art mentioned above) is material prior art that could have been used to form the basis for a rejection of the claims and that was not disclosed.

25. Smartner/SNIO is therefore entitled to a declaratory judgment that the claims of the '606 patent are unenforceable.

**COUNT IV**

**Declaratory Judgment of Noninfringement of the '679 Patent**

26. Smartner/SNIO repeats and realleges paragraphs 1 through 25 of this Complaint as if the same were full set forth herein.

27. Smartner/SNIO's products do not infringe any valid claim of the '679 patent, either directly, indirectly, contributorily, or otherwise. Smartner/SNIO has not induced others to infringe the '679 patent.

28. Smartner/SNIO is therefore entitled to a declaratory judgment that it does not infringe the '679 patent.

**COUNT V**

**Declaratory Judgment of Invalidity of the '679 Patent**

29. Smartner/SNIO repeats and realleges paragraphs 1 through 28 of this Complaint as if the same were full set forth herein.

30. The claims of the '679 patent are invalid for failure to meet the requirements specified in Title 35 of the United States Code, including, but not limited to, 35 U.S.C. §§ 101, 102, 103, and 112.

31. Smartner/SNIO is therefore entitled to a declaratory judgment that the '679 patent is invalid.

**COUNT VI**

**Declaratory Judgment of Unenforceability of the '679 Patent**

32. Smartner/SNIO repeats and realleges paragraphs 1 through 31 of this Complaint as if the same were full set forth herein.

33. The claims of the '679 patent are unenforceable by reason of their having been procured through inequitable conduct and fraud. Particularly, the applicants failed to advise the Examiner for the '679 patent that the '708 patent, the '590 patent, the '131 patent, and the '116 patent, patents of which the '679 claims priority, had been rejected, and the bases for the prior rejection by the Examiners for those patents. As such, the applicants knew of and were guilty of intentionally

concealing material information from the USPTO concerning the prosecution histories of the '708, '590, '131, and '116 patents. The rejected claims of the '708, '590, '131, and '116 applications and the claims of the '679 application are similar claims in a similar technology. On October 15, 1998, the USPTO rejected all of the claims in the '708 application in light of various pieces of prior art. Those pieces of prior art included Smith, Lagarde, Cardinal, and Cary. On February 24, 1998, the USPTO rejected all the claims in the '590 application in light of Brunson. On April 14, 1998, the USPTO rejected all of the claims in the '116 application in light of various pieces of prior art. Those pieces of prior art included Foley and Using Netscape 2. The USPTO further rejected all of the claims in the '116 application on October 27, 1998, in light of Weber and Gish, and on April 12, 1999, in light of Cowan. On January 25, 1999, the USPTO rejected claims 28-46 of the '131 application in light of Buchanan. On January 21, 2005, the attorney of record for the '679 patent submitted 14 IDS's encompassing 208 pieces of art to the USPTO. The IDS's listed the Smith, Lagarde, Cardinal, Cary, Foley, Brunson, and Buchanan art. However, the attorney of record failed to apprise the Examiner that the cited art was used in the rejection of the '708, '116, '131, and '590 applications. Therefore, Visto had knowledge of information material to the patentability of the '679 application and failed to disclose that information to the USPTO.

34. Additionally, the applicants for the '679 patent were also guilty of other deceptions, concealments, and misrepresentations before the USPTO. Particularly, during the prosecution of the '679 patent, applicants failed to advise the Examiner that U.S. Patent No. 6,085,192 (the "'192" patent), a patent of which the '679 claims priority, had been rejected during reexamination, and the bases for the prior rejection by the Examiners. As such, the applicants knew of and were guilty of intentionally concealing material information from the USPTO concerning the prosecution histories of the '192 patent. The rejected claims of the '192 application and the claims of the '679 application are similar claims in a similar technology. On February 7, 2005, the USPTO rejected claims 1, 9-11, and 20-25 in the '192 application in light of various pieces of prior art. Those pieces of prior art included U.S. Patent No. 5,857,201 to Wright et al. ("Wright") and U.S. Patent No. 6,006,274 to Hawkins et al. ("Hawkins"). On January 21, 2005, the attorney of record for the '679 patent submitted 14 IDS's

encompassing 208 pieces of art to the USPTO. Although the IDS's listed the Wright and Hawkins art, the attorney of record failed to apprise the Examiner that the cited art was used in the rejection of the '192 application. Therefore, Visto had knowledge of information material to the patentability of the '679 application and failed to disclose that information to the USPTO.

35. Furthermore, the applicants for the '679 patent were also guilty of deceptions, concealments, and misrepresentations before the USPTO for failing to advise the Examiner that the request for reexamination of the 708 patent, a patent of which the '679 claims priority, had been granted. In response to the request for reexamination of the '708 patent, the Examiner concluded that a substantial question of patentability existed in light of U.S. Patent No. 5,727,202 to Kucala ("Kucala"). As such, the applicants knew of and were guilty of intentionally concealing material information from the USPTO concerning the grant of reexamination of the '708 patent. Visto had knowledge of information material to the patentability of the '679 application and failed to disclose that information to the USPTO.

36. Additional acts of inequitable conduct were committed during the prosecution of the '679 patent. Despite its continuing duty of disclosure, the applicants never directed the Examiner to the relevant features of Lotus Notes (*e.g.*, "replication"). Instead, Lotus Notes references were merely included in a long list of prior art, without directing the Examiner to any relevant aspect of them.

37. Two references (K. Brown, et al., *Mastering Lotus Notes* published by Cybex Inc. (1995); P. Grous, "Creating and Managing a Web Site with Lotus' InterNotes Web Publisher," *The View* Vol. 1, Issue 4 (September/October 1995)), produced to Visto on August 8, 2005, were not brought to the attention of the Examiner of the '679 patent. Failure to point out the relevant features of the Lotus Notes references and relevant patents is further evidence that the '679 patent is unenforceable due to Visto's inequitable conduct. The applicants failed to disclose the above material information or comply with M.P.E.P. § 2001.6(c) with an intent to deceive. Accordingly, the '679 patent is invalid.

38. Smartner/SNIO is therefore entitled to a declaratory judgment that the claims of the '679 patent are unenforceable.

## PRAYER FOR RELIEF

WHEREFORE, Plaintiff Seven Networks International OY prays that the Court enter judgment that:

a) U.S. Patent No. 6,151,606 is not infringed by Smartner/SNIO's products;

b) The claims of U.S. Patent No. 6,1515,606 are invalid;

c) U.S. Patent No. 6,151,606 is unenforceable;

d) U.S. Patent No. 7,039,679 is not infringed by Smartner/SNIO's products;

e) The claims of U.S. Patent No. 7,039,679 are invalid; and

f) U.S. Patent No. 7,039,679 is unenforceable.

Dated: February 6, 2007

Respectfully submitted,

HOWREY LLP

By: _____

Henry C. Bunsow
K.T. Cherian
James C. Pistorino
James F. Valentine

Attorneys for Plaintiff SEVEN NETWORKS INTERNATIONAL OY

8433026

# EXHIBIT A

US006151606A

# United States Patent [19]

**Mendez**

[11] **Patent Number: 6,151,606**

[45] **Date of Patent: Nov. 21, 2000**

[54] **SYSTEM AND METHOD FOR USING A WORKSPACE DATA MANAGER TO ACCESS, MANIPULATE AND SYNCHRONIZE NETWORK DATA**

[75] Inventor: **Daniel J. Mendez**, Menlo Park, Calif.

[73] Assignee: **Visto Corporation**, Mountain View, Calif.

[21] Appl. No.: **09/008,354**

[22] Filed: **Jan. 16, 1998**

[51] **Int. Cl.**[7] ..... **G06F 17/30**

[52] **U.S. Cl.** ..... **707/201**; 707/8; 707/10; 707/202; 707/203

[58] **Field of Search** ..... 707/8, 10, 202, 707/203, 506, 511; 709/103, 201, 204, 224, 228, 302, 303; 706/14, 45; 345/302, 340; 320/257, 463; 705/35; 395/500.32

[56] **References Cited**

U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 4,831,582 | 5/1989 | Miller et al. | 707/104 |
| 4,875,159 | 10/1989 | Cary et al. | 364/200 |
| 4,897,781 | 1/1990 | Chang | 364/200 |
| 5,263,157 | 11/1993 | Janis | 707/9 |
| 5,386,564 | 1/1995 | Shearer et al. | 707/500 |
| 5,392,390 | 2/1995 | Crozier | 395/161 |
| 5,572,643 | 11/1996 | Judson | 395/793 |
| 5,581,749 | 12/1996 | Hossain et al. | 707/1 |
| 5,600,834 | 2/1997 | Howard | 395/617 |
| 5,613,012 | 3/1997 | Hoffman et al. | 382/115 |
| 5,623,601 | 4/1997 | Vu | 395/187.01 |
| 5,627,658 | 5/1997 | Connors et al. | 358/407 |
| 5,634,053 | 5/1997 | Noble et al. | 707/4 |
| 5,647,002 | 7/1997 | Brunson | 380/49 |
| 5,652,884 | 7/1997 | Palevich | 395/651 |
| 5,666,530 | 9/1997 | Clark et al. | 395/617 |
| 5,666,553 | 9/1997 | Crozier | 395/803 |
| 5,678,039 | 10/1997 | Hinks et al. | 395/604 |
| 5,680,542 | 10/1997 | Mulchandani | 395/183.04 |
| 5,682,524 | 10/1997 | Freund et al. | 395/605 |
| 5,684,990 | 11/1997 | Boothby | 707/203 |
| 5,687,322 | 11/1997 | Deaton | 705/14 |
| 5,701,400 | 12/1997 | Amado | 706/45 |
| 5,701,423 | 12/1997 | Crozier | 395/335 |
| 5,706,502 | 1/1998 | Foley et al. | 707/10 |

(List continued on next page.)

OTHER PUBLICATIONS

Article by Bellovin et al., entitled: "Network Firewalls" Published by IEEE Communications Magazine Sep. 1994, pp. 50–57.

Article by Steffen Stempel, entitled: "IPAccess—An Internet Service Access System for Firewall Installations" Published by IEEE Communications Magazine Feb. 16, 1995, pp. 31–41.

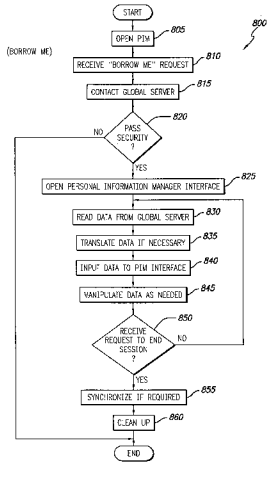
(List continued on next page.)

*Primary Examiner*—Thomas G. Black
*Assistant Examiner*—Diane D. Mizrahi
*Attorney, Agent, or Firm*—Graham & James LLP

[57] **ABSTRACT**

A system includes a communications module for downloading workspace data from a remote site, an application program interface coupled to the communications module for communicating with a workspace data manager to enable manipulation of the downloaded workspace data and thereby create manipulated data, and a general synchronization module coupled to the communications module for synchronizing the manipulated data with the workspace data stored at the remote site. An instantiator requests the workspace data manager to provide an interface for enabling manipulation of the downloaded workspace data. The workspace data manager may create another instance of the interface or may provide access to its only interface to enable manipulation of the data. A data reader may translate the downloaded workspace data from the format used by the remote site to the format used by the workspace data manager. Upon logout, a de-instantiator synchronizes the data with the global server and deletes the workspace data. The system handles the situation where the data stored at the remote site has not changed and therefore includes the downloaded data, and the situation the data stored at the remote site has been modified and therefore is different than the downloaded data.

**21 Claims, 6 Drawing Sheets**



U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 5,710,918 | 1/1998 | Lagarde et al. | 395/610 |
| 5,713,019 | 1/1998 | Keaten | 395/610 |
| 5,715,403 | 2/1998 | Stefik | 705/44 |
| 5,717,925 | 2/1998 | Harper et al. | 395/613 |
| 5,721,908 | 2/1998 | Lagarde et al. | 395/610 |
| 5,721,914 | 2/1998 | DeVries | 707/104 |
| 5,729,735 | 3/1998 | Meyering | 395/610 |
| 5,745,360 | 4/1998 | Leone et al. | 364/140 |
| 5,757,916 | 5/1998 | MacDoran et al. | 380/25 |
| 5,758,150 | 5/1998 | Bell et al. | 395/610 |
| 5,758,354 | 5/1998 | Huang et al. | 707/201 |
| 5,758,355 | 5/1998 | Buchanan | 707/201 |
| 5,765,171 | 6/1998 | Gehani et al. | 707/203 |
| 5,778,346 | 7/1998 | Frid-Nielsen et al. | 395/208 |
| 5,790,425 | 8/1998 | Wagle | 364/551.01 |
| 5,790,790 | 8/1998 | Smith et al. | 395/200.36 |
| 5,799,318 | 8/1998 | Cardinal et al. | 707/104 |
| 5,832,483 | 11/1998 | Barker | 707/8 |
| 5,862,325 | 1/1999 | Reed | 395/200.31 |
| 5,870,759 | 2/1999 | Bauer et al. | 707/201 |
| 5,951,652 | 9/1999 | Ingrassia et al. | 709/248 |
| 5,966,714 | 10/1999 | Huang et al. | 707/201 |
| 5,999,947 | 12/1999 | Zollinger et al. | 707/203 |

OTHER PUBLICATIONS

Article by Braun et al., entitled: "Web Traffic Characterization: an assessment of the impact of caching documents from NCSA's web server" Published by Elsevier Science B.V. 1995 pp. 37–51.

Article by Nelson et al., entitled: "Security for Infinite Networks" Published by IEEE Communications Magazine on Aug. 22, 1995, pp. 11–19.

Article by Greenwald et al., entitled: "Designing an Academic Firewall: Policy, Pratice, and Experience with SURF" Published by IEEE Communications Magazine on Feb. 22, 1996, pp. 79–92.

Article by Kiuchi et al., entitled: "C–HTTP—The Development of a Secure, Closed HTTP–based Network on the Internet" Published by IEEE Proceedings of SNDSS on Feb. 22, 1996, pp. 64–75.

Article by S. Cobb, entitled: "Establishing Firewall Policy" Published by National Computer Security Assn. on Jun. 25–27, 1996, pp. 198–205.

Margaret J. Brown, "The Visto Briefcase Pro Puts Your PIM On The Internet", URL:http://www.zdnet.com/zdnn/stories/zdnn_display/0,3440,341892,00.html, Aug. 13, 1998, 1 page.

Web site entitled "Bookmark Translator 2.0: This Utility transform Microsoft Internet Explore's bookmarks in the format valid for Netscape Navigator and viceversa," Enzo Marinacci, Rome–Jul. 1997, URL=http://www.bns.it/emware/BookmarkTranslator–uk.htm, pp. 1–4.

FIG. 1
(NETWORK SYSTEM)

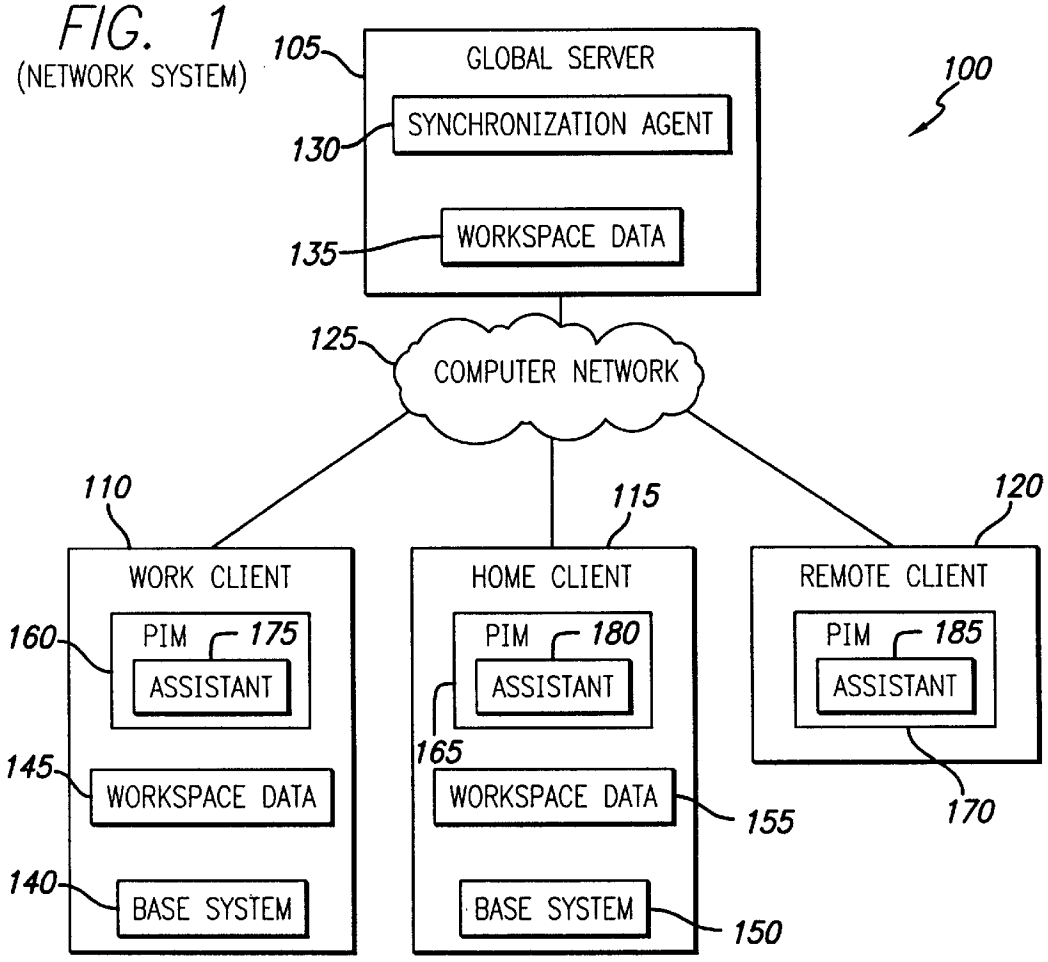
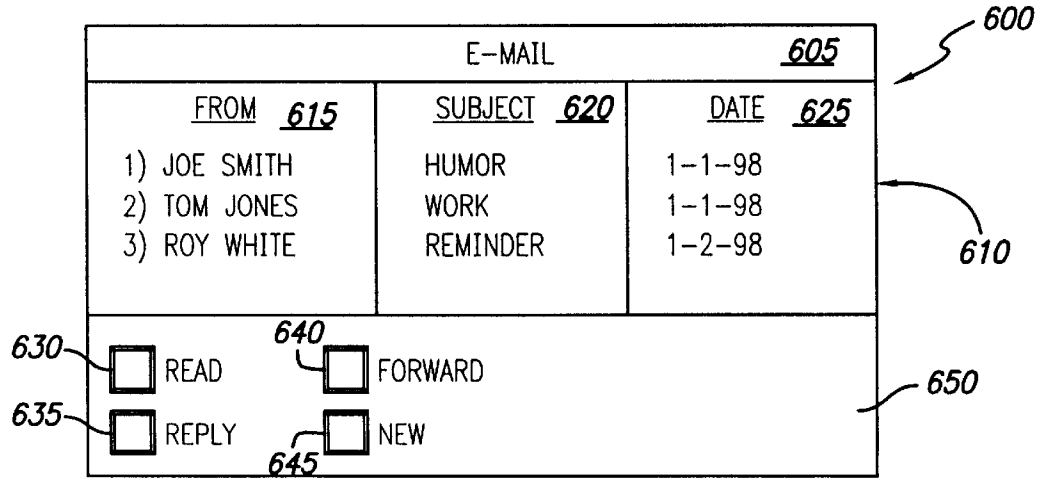



FIG. 6



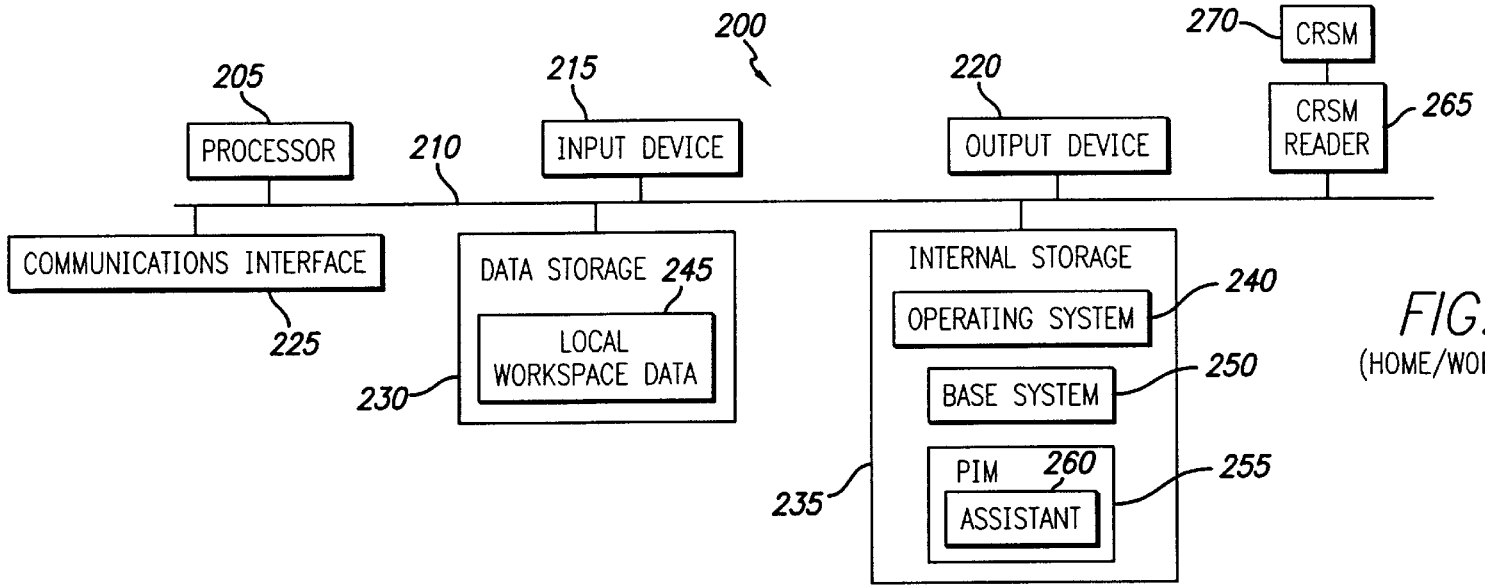



FIG. 2
(HOME/WORK CLIENT)

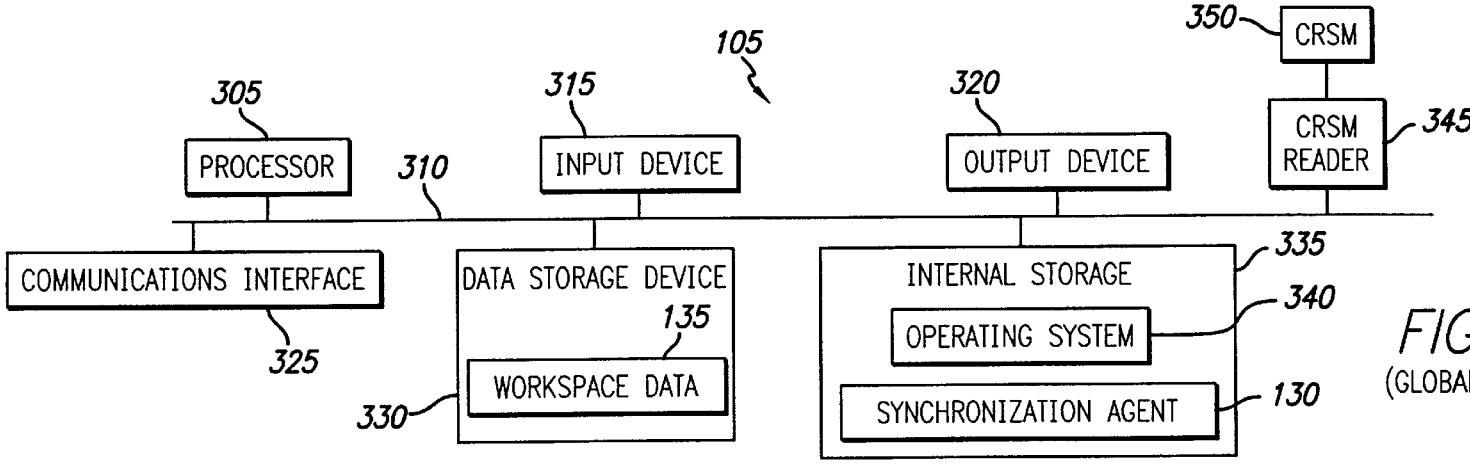



FIG. 3
(GLOBAL SERVER)

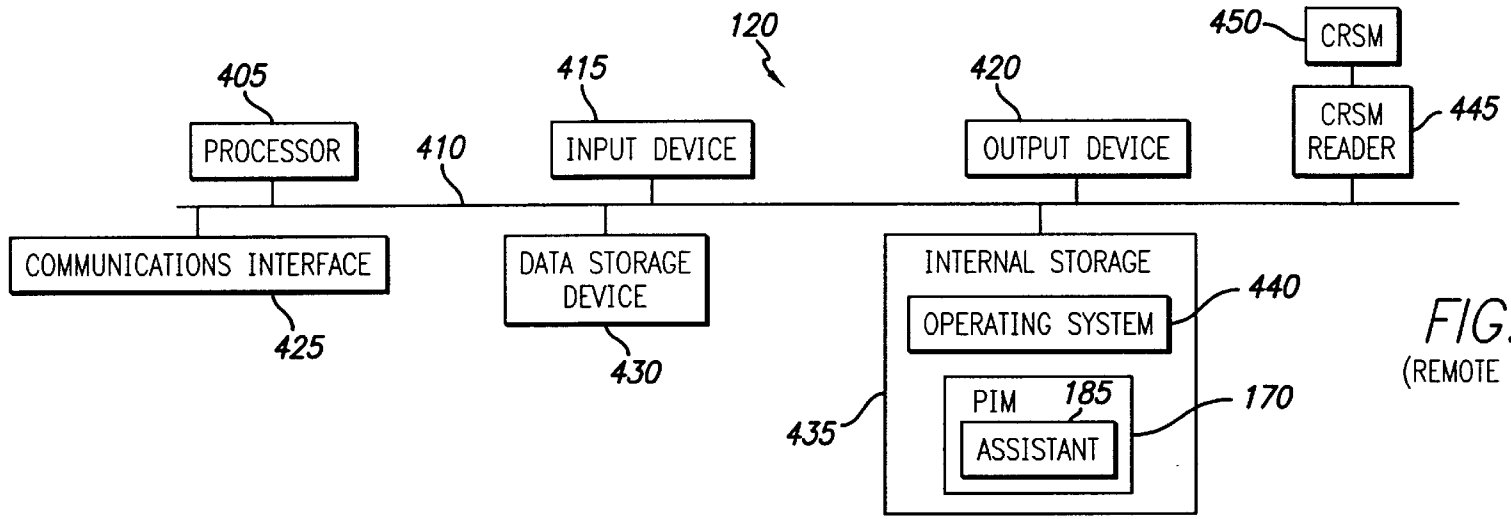



FIG. 4
(REMOTE CLIENT)

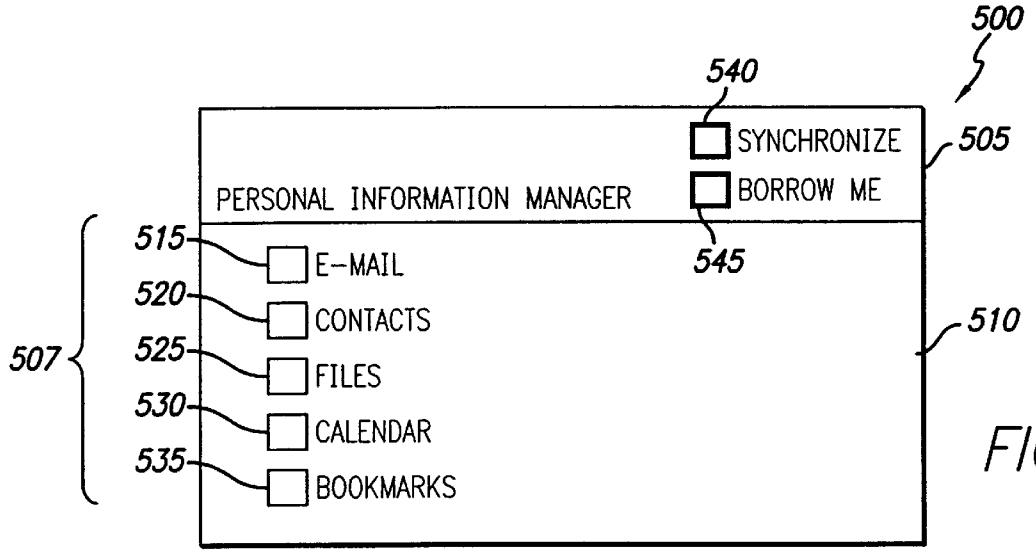



FIG. 5

FIG. 7

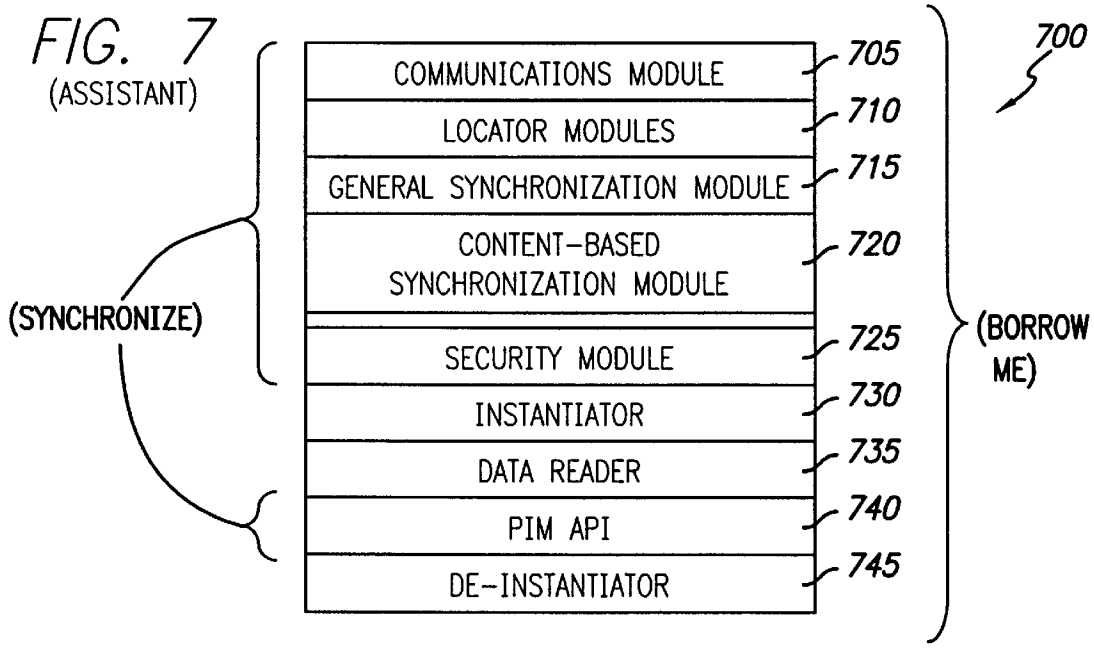
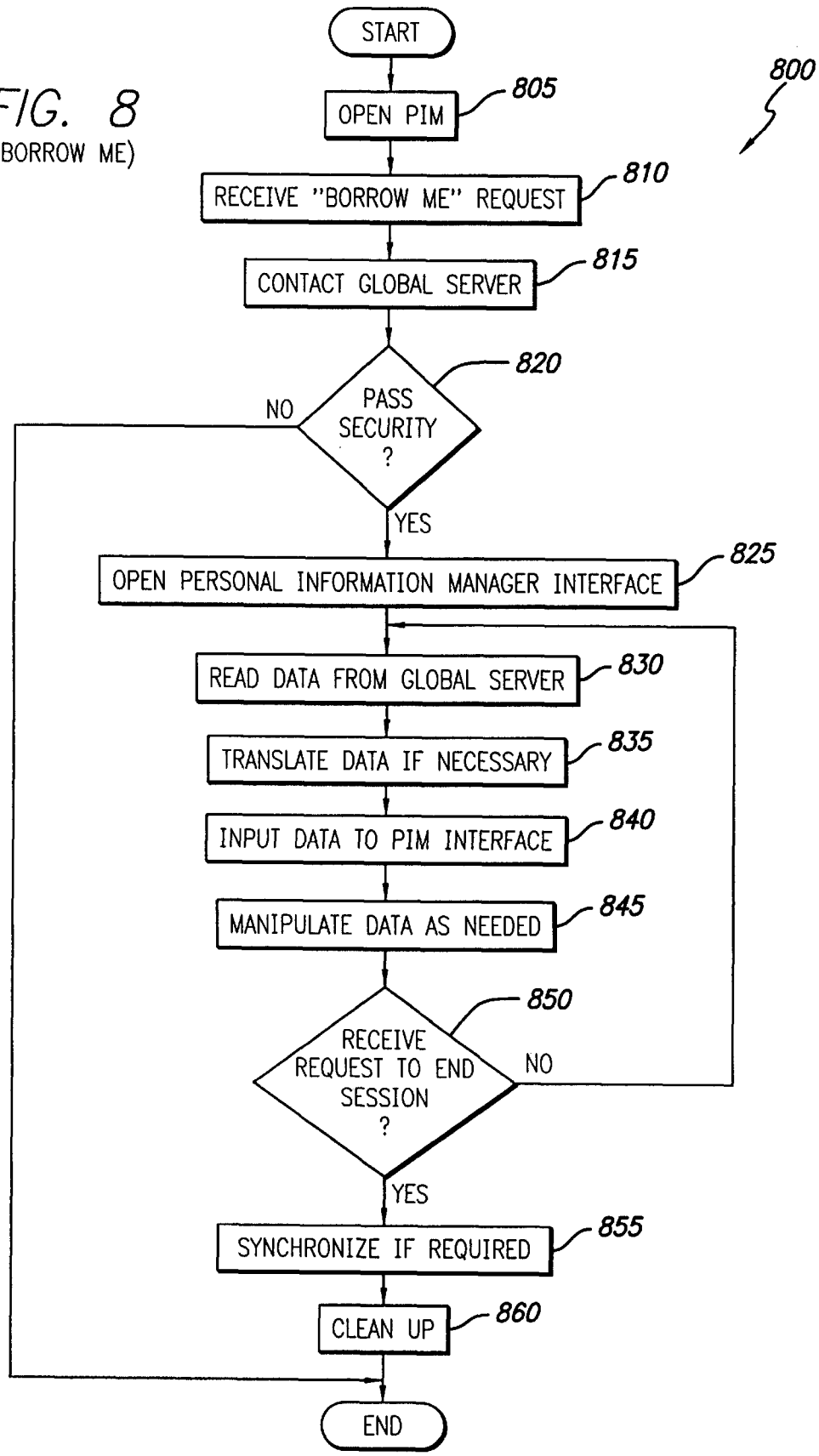

(ASSISTANT)
(SYNCHRONIZE)
COMMUNICATIONS MODULE
LOCATOR MODULES
GENERAL SYNCHRONIZATION MODULE
CONTENT-BASED SYNCHRONIZATION MODULE
SECURITY MODULE
INSTANTIATOR
DATA READER
PIM API
DE-INSTANTIATOR
705
710
715
720
725
730
735
740
745
700
(BORROW ME)

FIG. 8
(BORROW ME)



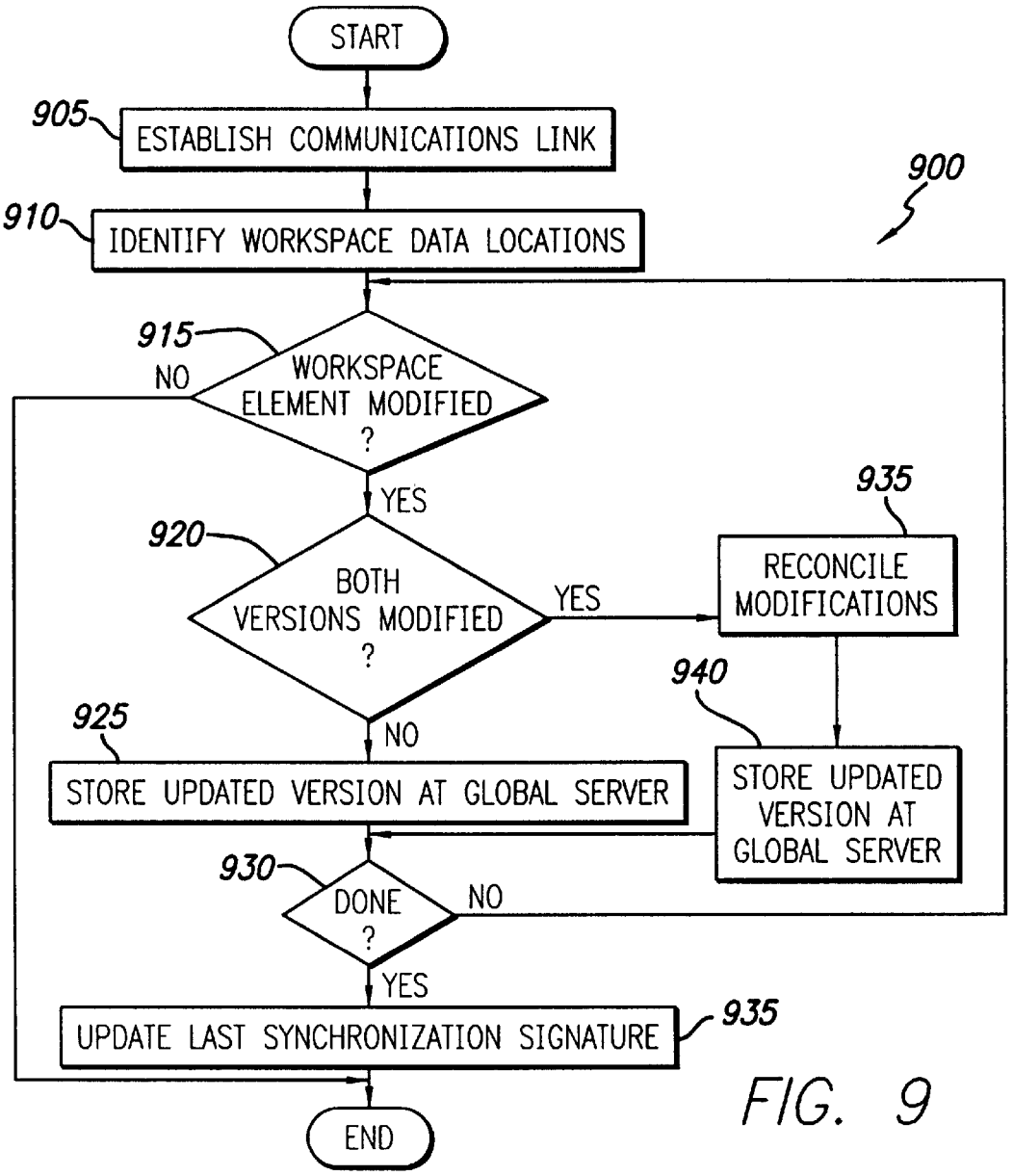

START
905
ESTABLISH COMMUNICATIONS LINK
900
910
IDENTIFY WORKSPACE DATA LOCATIONS
915
NO
WORKSPACE ELEMENT MODIFIED ?
YES
935
920
BOTH VERSIONS MODIFIED ?
YES
RECONCILE MODIFICATIONS
940
925
NO
STORE UPDATED VERSION AT GLOBAL SERVER
STORE UPDATED VERSION AT GLOBAL SERVER
930
DONE ?
NO
YES
UPDATE LAST SYNCHRONIZATION SIGNATURE
935
END

FIG. 9

# SYSTEM AND METHOD FOR USING A WORKSPACE DATA MANAGER TO ACCESS, MANIPULATE AND SYNCHRONIZE NETWORK DATA

## PRIORITY REFERENCES

This application claims priority of and hereby incorporates by reference U.S. patent application Ser. No. 08/766,307 pending, entitled "System and Method for Globally Accessing Computer Services," filed on Dec. 13, 1996, by inventors Mark D. Riggins, et al; U.S. patent application Ser. No. 08/841,950 pending, entitled "System and Method for Enabling Secure Access to Services in a Computer Network, filed on Apr. 8, 1997, by inventor Mark D. Riggins; U.S. patent application Ser. No. 08/865,075, and now U.S. Pat. No. 6,023,708 entitled "System and Method for Using a Global Translator to Synchronize Workspace Elements Across a Network," filed on May 29, 1997, by inventors Daniel J. Mendez, et al.; U.S. patent application Ser. No. 08/835,997 pending, entitled "System and Method for Securely Synchronizing Multiple Copies of a Workspace Element in a Network," filed on Apr. 11, 1997, by inventors Daniel J. Mendez, et al.; U.S. patent application Ser. No. 08/897,888 pending and now U.S. Pat. No. 5,961,590, entitled "System and Method for Synchronizing Electronic Mail Across a Network," filed on Jul. 22, 1997, by inventors Daniel J. Mendez, et al.; U.S. patent application Ser. No. 08/899,277, entitled "System and Method for Using an Authentication Applet to Identify and Authenticate a User in a Computer Network," filed on Jul. 23, 1997, by inventor Mark D. Riggins; and U.S. patent application Ser. No. 08/903,118 pending, entitled "System and Method for Globally and Securely Accessing Unified Information in a Computer Network," filed on Jul. 30, 1997, by inventors Daniel J. Mendez, et al.

## BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly provides a system and method for using a workspace data manager to access network data.

2. Description of the Background Art

Data accessibility and consistency are significant concerns for computer users. For example, when a roaming user, i.e., a user who travels to a remote location, needs to review or manipulate data such as an e-mail or prepared document, the roaming user must either carry the data to the remote location or access a workstation remotely. Maintaining a true copy of a database is a cumbersome process. Accordingly, system designers have developed an array of techniques for connecting a remote terminal across a computer network to the workstation storing the data.

To guarantee readability of the downloaded data, the user must carry a laptop computer containing all the applications needed to present and enable manipulation of the downloaded data, or find a network-connected computer that contains the needed application programs. Further, when maintaining multiple independently modifiable copies of particular data, a user risks using an outdated version. By the time the user notices an inconsistency, interparty miscommunication or data loss may already have resulted. The user must then spend more time reconciling the inconsistent versions.

The problems of data accessibility and inconsistency are exacerbated when multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.



## SUMMARY OF THE INVENTION

The present invention provides a system for using a workspace data manager to access, manipulate and synchronize workspace data. A workspace data manager may include a Personal Information Manager (PIM), a word processing program, a spreadsheet program, or any application program that enables manipulation of workspace data. Workspace data includes at least one workspace element, such as an e-mail, a day of calendar data, a word document, a bookmark, a sheet of spreadsheet data, or a portion thereof. Workspace data may include e-mails, calendar data, word documents, bookmarks, spreadsheet data, or portions thereof

The system includes a communications module for downloading workspace data from a remote site, an application program interface coupled to the communications module for communicating with a workspace data manager to enable manipulation of the downloaded workspace data and thereby create manipulated data, and a general synchronization module coupled to the communications module for synchronizing the manipulated data with the workspace data stored at the remote site. An instantiator requests the workspace data manager to provide an interface for enabling manipulation of the downloaded workspace data. The workspace data manager may create another instance of the interface or may provide access to its only interface to enable manipulation of the data. A data reader translates the downloaded workspace data from the format used by the remote site to the format used by the workspace data manager. For example, data stored at the global server site in a canonical format may be translated to Organizer™, Outlook™ or other workspace element manager format. Upon logout, a de-instantiator initiates synchronization and deletes the data stored locally. It will be appreciated that the system handles the situation where the data stored at the remote site has not changed and therefore includes the downloaded data, and the situation the data stored at the remote site has been modified and therefore is different than the downloaded data.

The present invention further provides a method of using a workspace data manager to enable access, manipulate and synchronize workspace data. The method comprises the steps of downloading data from a remote site, requesting a workspace data manager to enable manipulation of the data and thereby create manipulated data, and synchronizing the manipulated data with the data stored at the remote site.

The system and method of the present invention advantageously enable the use of an integral interface, instead of using an interface for the synchronization software, an interface for the workspace data manager and an interface for the communication engine downloading the workspace data. Accordingly, the user need not become familiar with multiple interfaces. The user need only find a remote site that includes a workspace data manager that includes assistant-like functionality. Assistant-like functionality includes services for interfacing between the workspace data manager and the global server. Because the system and method substitute the global data for the local data, or create an instance for the global data, the system and method further advantageously enable a workspace data manager to provide

an interface for manipulating workspace data without compromising the local data.

Further, the system and method advantageously provide a simple graphical user interface for enabling borrowing of the workspace data manager and synchronization of manipulated data. The system and method also advantageously delete downloaded data and all interfaces from the local client, so that no traces are left on the local client for unprivileged users to review. Using the technology described in the applications incorporated by reference above, the system and method of the present invention further enable access and synchronization of data across different workspace data manager formats and across network firewalls.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating a network system, in accordance with the present invention;

FIG. **2** is a block diagram illustrating details of the home or work client of FIG. **1**;

FIG. **3** is a block diagram illustrating details of the global server of FIG. **1**;

FIG. **4** is a block diagram illustrating details of the remote client of FIG. **1**;

FIG. **5** is a block diagram illustrating details of an assistant of FIG. **1**;

FIG. **6** illustrates a personal information manager interface;

FIG. **7** illustrates a second personal information manager interface incorporating an assistant interface;

FIG. **8** is a flowchart illustrating a method of accessing network data from a remote site in accordance with the present invention; and

FIG. **9** is a flowchart illustrating a method of synchronizing network data from a remote site.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. **1** is a block diagram illustrating a network system **100** for using a workspace data manager to access, manipulate and synchronize workspace data in accordance with the present invention. A workspace data manager may include a Personal Information Manager (PIM), a word processing program, a spreadsheet program, or any application program that enables manipulation of workspace data. Workspace data includes at least one workspace element, such as an e-mail, a day of calendar data, a word document, a bookmark, a sheet of spreadsheet data, or a portion thereof. Workspace data may include e-mails, calendar data, word documents, bookmarks, spreadsheet data, or portions thereof. Although the network system **100** is described with reference to PIM's, one skilled in the art will recognize that the system **100** will work with any workspace data manager.

Network system **100** includes a global server **105** coupled via a computer network **125** to a work client **110**, to a home client **115** and to a remote client **120**. The global server **105** includes a synchronization agent **130** and workspace data **135**. The work client **110** includes a base system **140** and workspace data **145**. The home client **115** includes a base system **150** and workspace data **155**.

Each of the base system **140** and the base system **150** cooperate with the synchronization agent **130** to synchronize workspace data **135**, workspace data **145** and workspace data **155** between the work client **110**, the home client **115** and the global server **105**. Synchronization of workspace data **135**, **145** and **155** is described in detail in the patent applications incorporated by reference above. However, a brief example of synchronization is provided for completeness.

First, the base system **140** on the work client **110** site negotiates a secure communications channel via any firewalls with the synchronization agent **130**, for example, using Secure Sockets Layer (SSL) technology. The base systems **140** examines version information and if necessary the content of a workspace to determine the most updated version. The most updated version is then stored at the client **110** site and at the global server **105** site. The base system **140** repeats these operations for all workspace elements selected for synchronization. Second, the base system **150** on the home client **115** site uses similar steps to synchronize its workspace data **155** with the workspace data **135** on the global server **105** site. Accordingly, the most updated versions of the workspace data **135**, **140** and **145** are stored at all three sites.

Each of the work client **110**, the home client **115** and the remote client **120** includes a respective workspace data manager, e.g., a Personal Information Manager (PIM) **160**, **165** and **170** such as Outlook™ 98 developed by Microsoft Corporation, Organizer 97 developed by Lotus Development Corporation or Sidekick 98 developed by Starfish Software. Each PIM **160**, **165** and **170** includes an assistant **175**, **180** and **185** that adds data access and synchronization finctions to the PIM **160**, **165** and **170**. Accordingly, a user can transparently use an assistant **175**, **180** or **185** via a PIM **160**, **165** or **170** to access workspace data **135** from the global server **105**, to present and enable manipulation of downloaded workspace data **135**, and to synchronize manipulated downloaded data **135** with the workspace data **135** stored on the global server **105**. Components and operations of the assistant **175**, **180** or **185** are described in detail with reference to FIGS. **7–9**.

FIG. **2** is a block diagram illustrating details of a data-synchronizing client **200**, in a generic embodiment which exemplifies each of the work client **110** and the home client **115**. The client **200** includes a processor **205**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel **210**. The client **200** further includes an input device **215** such as a keyboard and mouse, an output device **220** such as a Cathode Ray Tube (CRT) display, data storage **230** such as a magnetic disk, and internal storage **235** such as Random-Access Memory (RAM), each coupled to the communications channel **210**. A communications interface **225** couples the communications channel **210** to the computer network **125**.

An operating system **240** controls processing by processor **205**, and is typically stored in data storage **230** and loaded into internal storage **235** (as illustrated) for execution. A base system **250**, which cooperates with the synchronization agent **130** for synchronizing local workspace data **245** with workspace data **135**, also may be stored in data storage **230** and loaded into internal storage **235** (as illustrated) for execution by processor **205**. The local workspace data **245** exemplifies workspace data **145** or workspace data **150**, and may be stored in data storage **230**.

A PIM **255** includes an assistant **260**, which enables a user to download workspace data **135** from the global server **105**, and to use the PIM **255** for displaying and manipulating the workspace data **135**. The assistant **260** further enables the PIM **255** to synchronize the manipulated data **135** with the

workspace data **135** on the global server **105**. The PIM **255** exemplifies each of the PIM **160** on the work client **110** and the PIM **165** on the home client **115**. The assistant **260** exemplifies each of the assistant **175** on the work client **110** and the assistant **180** on the home client **115**. The PIM **255** may be stored in data storage **230**, and loaded into internal storage **235** (as illustrated) for execution by the processor **205**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a computer-readable storage medium (CRSM) reader **265** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **210** for reading a computer-readable storage medium (CRSM) **270** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via the CRSM reader **265**.

FIG. **3** is a block diagram illustrating details of the global server **105**. The global server **105** includes a processor **305**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel **310**. The global server **105** further includes an input device **315** such as a keyboard and mouse, an output device **320** such as a CRT display, data storage **325** such as a magnetic disk, and internal storage **330** such as RAM, each coupled to the communications channel **310**. A communications interface **325** couples the communications channel **310** to the computer network **125**.

An operating system **340** controls processing by processor **305**, and is typically stored in data storage **330** and loaded into internal storage **335** (as illustrated) for execution. The synchronization agent **130**, which cooperates with the base system **250** (FIG. **2**) for synchronizing local workspace data **245** with workspace data **135**, also may be stored in data storage **330** and loaded into internal storage **335** (as illustrated) for execution by processor **305**. The workspace data **135** may be stored in data storage **230**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a CRSM reader **345** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **310** for reading a CRSM **350** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via the CRSM reader **345**.

FIG. **4** is a block diagram illustrating details of the remote client **120**. The client **120** includes a processor **405**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel **410**. The client **120** further includes an input device **415** such as a keyboard and mouse, an output device **420** such as a CRT display, data storage **425** such as a magnetic disk, and internal storage **430** such as RAM, each coupled to the communications channel **410**. A communications interface **425** couples the communications channel **410** to the computer network **125**.

An operating system **440** controls processing by processor **405**, and is typically stored in data storage **430** and loaded into internal storage **435** (as illustrated) for execution. The PIM **170** and assistant **185** may be stored in data storage **430**, and loaded into internal storage **435** (as illustrated) for execution by the processor **405**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a CRSM reader **445** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **310** for reading a CRSM **450** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via the CRSM reader **445**.

FIG. **5** is a block diagram illustrating a PIM interface **500**, which includes a header **505** and a selection window **510**.

The header **505** includes a synchronize button **540** and a "borrow me" button **545**, which are presented by the assistant **175**, **180** or **185** incorporated in the PIM **160**, **165** or **170**. Invoking the synchronize button **540** causes the assistant **175**, **180** or **185** to enable synchronization of data entered into the PIM **160**, **165** or **170** with the workspace data **135** on the global server **135**. The synchronize button **540** may enable the user to configure a preference file that indicates when automatic synchronization is to initiate and may also enable a user to effect manual synchronization.

The "borrow me" button **545** enables a user to use a PIM **160**, **165** or **170** for viewing and manipulating workspace data **135** downloaded from the global server **105**. That is, invoking the "borrow me" button **545** causes the corresponding assistant **175**, **180** or **185** to communicate with the global server **105**, to provide user identification and authentication information to the global server **105**, to download workspace data **135** from the global server **105**, to display and enable manipulation of the downloaded data **135** using the PIM interface **500**, and to synchronize the manipulated downloaded data **135** upon logout. Since the PIM interface **500** is provided by the pre-existing PIM, the assistant **175**, **180** or **185** need not provide its own data interface. Only a single interface is needed.

It will be appreciated that upon logout, the base systems **140** and **150** will cooperate with the synchronization agent **130** to synchronize automatically the workspace data **135** on the global server **105** with the workspace data **145** and **155**. Accordingly, the user always has access to the most updated versions of workspace data from any site that executes a PIM **160**, **165** or **170** having an assistant **175**, **180** or **185** embodied therein.

It will be appreciated that the synchronize button **540** is most helpful to the work client **110** and the home client **115**, since typically the work client **110** and home client **115** will set the preference file to configure automatic synchronization. Synchronization of the manipulated workspace data **135** at the remote client **120** will most often be effected through the automatic logout procedures of the "borrow me" button. Logout is described in greater detail with reference to the Outlook™ and Lotus Organizer examples shown and described below with reference to FIG. **7**. Accordingly, the borrow me button **545** is most helpful to the remote client **120**.

The selection window **510** provides a list of buttons **507**, wherein each button **507** corresponds to a set of workspace elements, e.g., e-mails **515**, contacts **520**, files **525**, calendar data **530** and bookmarks **535**. A mouse-down on a virtual button **507** causes the selection of a corresponding workspace element set and the selection of a corresponding user interface for displaying and enabling manipulation of the workspace elements included in the set. For example, selection of button **515** selects the e-mail set, and selects a corresponding user interface for displaying, writing, forwarding, etc. e-mails. Selecting a button **507** causes the assistant **175**, **180** or **185** to download the corresponding workspace data **135**, and causes the PIM **160**, **165** or **170** to display and enable manipulation of the downloaded data **135** on a workspace element set interface (shown and described with reference to FIG. **6**).

FIG. **6** illustrates an example e-mail workspace element set user interface **600** (commonly referred to as the "In-Box") for displaying received e-mails. The user interface **600** includes a header **605**, an e-mail list window **610** and a manipulation command window **650**.

The header **605** lists the name of the workspace element set, namely, "E-Mail." The e-mail list window **610** comprises three columns, including an origin column **615** which provides the origin of each e-mail, a subject column **620** which provides the subject of each e-mail, and a date column **625** which provides the date each e-mail was received. The e-mail list window **610** may display e-mails stored in a local e-mail database (not shown), e-mails stored in the e-mail server (not shown) or e-mails downloaded from the global server **105**. The e-mails shown include a first e-mail from Joe Smith, a second e-mail from Tom Jones, and a third e-mail from Roy White. If the user depressed the "borrow me" button **545** shown in FIG. **5**, then the e-mail list displayed would be the e-mails stored and downloaded from the global server **105**.

The manipulation window **650** includes available functions such as the conventional e-mail read function **630**, e-mail reply function **635**, e-mail forward function **640** and new e-mail write function **645**. It will be appreciated that the columns and functions will vary based on the PIM.

FIG. **7** is a block diagram illustrating details of a generic assistant **700**, which exemplifies each of the assistant **175**, **180** and **185**. The generic assistant **700** includes a communications module **705**, locator modules **710**, a general synchronization module **715**, a content-based synchronization module **720**, a security module **725**, an instantiator **730**, a data reader **735**, a PIM Application Program Interface (API) **740** and a de-instantiator **745**. The synchronization function of the assistant **700** uses the communications module **705**, the locator modules **710**, the general synchronization module **715**, the content-based synchronization module **720**, the security module **725** and the PIM API **740**. The "borrow me" function of the assistant **700** uses the communications module **705**, the locator modules **710**, the security module **725**, the instantiator **730**, the data reader **735**, the PIM API **740** and the de-instantiator **745**.

The communications module **705** includes routines for compressing and decompressing data, and routines for communicating with the synchronization agent **130**. The communications module **705** may apply Secure Socket Layer (SSL) technology to establish a secure communication channel. Examples of communications modules **705** may include TCP/IP stacks or the AppleTalk protocol.

The locator modules **710** include routines for identifying the memory locations of the workspace elements in the workspace data **135**. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue boxes to query the user. Accordingly, the locator modules **710** determine the memory addresses of the workspace elements in e-mail workspace data **135**, in file workspace data **135**, in calendar workspace data **135**, etc.

The general synchronization module **715** examines the workspace data **135** on the global server **105** to determine whether it had been modified while the user manipulated the data on the client **110**, **115** or **120**. Further, the general synchronization module **715** determines whether the user manipulated any data on the client **110**, **115** or **120**. If the general synchronization module **715** determines that only the data on the client **110**, **115** or **120** was manipulated, then the general synchronization module **715** computes and sends the changes to the synchronization agent **130** of the global server **105**. The general synchronization module **715** is initiated when the synchronization button **540** is depressed and during the logout procedures of the "borrow me" function.

The synchronization agent **130** then updates a last synchronization signature to indicate to all base systems **140** and **150** that synchronization with workspace data **145** and synchronization with workspace data **155** are needed. If the general synchronization module **715** determines that changes were made only to the workspace data **135** on the global server **105**, then the general synchronization module **715** instructs the synchronization agent **130** to compute and transmit the changes made to the client **110**, **115** or **120** at the client's request. The client **110** or **120** then updates its information. It will be appreciated that sending only the changes reduces processor load and increases transmission line efficiency, although alternatively an entire manipulated workspace element can be sent to the global server **105**.

If the general synchronization module **715** determines that the workspace data **135** on the global server **105** has been modified since download, and that the data on the client **110**, **115** or **120** has been modified, then the general synchronization module **715** instructs the content-based synchronization module **720** to perform its duties. The content-based synchronization module **720** includes routines for reconciling two or more modified versions of a workspace element. The content-based synchronization module **720** may request a user to select the preferred one of the modified versions or may respond based on preset preferences, i.e., by storing both versions in both stores or by integrating the changes into a single preferred version which replaces each modified version at both stores.

The security module **725** includes routines for obtaining user identification and authentication using such techniques as obtaining login and password information, obtaining a response to a challenge, obtaining a public key certificate, etc. The security module **725** performs identification and authentication techniques to confirm authorization by the user to access the workspace data **135** stored on the global server **105**. It will be appreciated that authorization may be granted only to portion of the workspace data **135** that belongs to the user.

The instantiator **730** is an application program interface **730** that creates a window for displaying and enabling manipulation of the workspace data **135** downloaded from the global server **105**. In an object-oriented environment, the instantiator **730** may create a new instance for the workspace data **135**. Alternatively, the instantiator **730** may store the

local data to a buffer (not shown) and use the current interface to display and enable manipulation of the workspace data **135**.

The data reader **735** communicates with the synchronization agent **130** at the global server **105**, and retrieves the workspace data **135** requested. For example, if the user depresses the "borrow me" button **545** (FIG. **5**) and depresses the e-mail button **515**, then the data reader **735** retrieves the e-mail workspace elements of the workspace data **135**, and delivers them to the PIM API **740**.

The PIM API **740** translates and transfers the workspace data **135** received from the global server **105** to the PIM **160**, **165** or **170** for display and enabling manipulation thereto. The PIM API **740** further translates and transfers the workspace data manipulated on the client **110**, **115** or **120** from the PIM **160**, **165** or **170** back to the global server **105**.

The de-instantiator **745** returns the PIM **160**, **165** or **170** to the state before the user selected the "borrow me" button **545**. The user may initiate operations of the de-instantiator **745** by depressing an "unborrow me" button (not shown) that is presented after selection of the "borrow me" button **545**. The de-instantiator **745** deletes any instance created by the instantiator **730**, deletes all workspace data **135** and data created by the user on the client **110**, **115** or **120** and automatically initiated synchronization of any manipulated downloaded data **135** with the workspace data **135** stored at the global server **105**.

Operations of the instantiator **730**, the data reader **735**, the PIM API **740** and the de-instantiator **745** are described in greater detail with reference to the following examples:

OUTLOOK EXAMPLE

| Action | Global Data | Local Data |
|---|---|---|
| standby | — | local data → $pst^{local}$ |
| button depressed | — | $pst^{local}$ |
| enter login/password | — | $pst^{local}$ |
| authenticate | — | $pst^{local}$ |
| send global data | global data → $pst^{local}$ | local data → $pst^{buffer}$ |
| manipulate data | global data → global data 2 | $pst^{buffer}$ |
| logout | 1) Compute Δglobal data<br>2) Synchronize Δglobal data with global server<br>3) Delete global data 2<br>4) | local data → $pst^{local}$ |

As illustrated by the Outlook™ example above, during standby, the PIM **160**, **165** or **170** stores the local data on the client **110**, **115** or **120** in a personal folder store $pst^{local}$. The user then depresses the "borrow me" button **545**. The security module **725** requests the user to enter a login and password, which the global server **105** authenticates. During these steps, it will be appreciated that the local data remains stored in $pst^{local}$. Upon user identification and authentication, the global server **105** sends the workspace data **135** (global data) to the requesting client **110**, **115** or **120**. The instantiator **730** on the client **110**, **115** or **120** transfers the local data from $pst^{local}$ to a buffer $pst^{buffer}$, and stores the received global data into $pst^{local}$. The data reader **745** and PIM API **740** enable the user to manipulate the global data, the manipulated data being referred to herein as "global data 2." Upon logout, for example, after an "unborrow me" button (not shown) is depressed, the global data 2 is synchronized with the workspace data **135**. Namely, the general synchronization module **715** determines the changes made (Δglobal data), and synchronizes Δglobal data with the workspace data **135**. The de-instantiator **745** deletes global data 2 and Δglobal data, and returns the local data to $pst^{local}$.

LOTUS ORGANIZER EXAMPLE

| Action | Global Data | Local Data |
|---|---|---|
| standby | — | local.org |
| button | — | local.org |
| enter login/password | — | local.org |
| authenticate | new instance | local.org |
| send global data | open with global.org | local.org |
| manipulate data | global.org → $global.org^2$ | local.org |
| logout | 1) compute Δglobal.org<br>2) Synchronize Δglobal.org with global server<br>3) delete $global.org^2$ | local.org |

As illustrated by the Lotus Organizer example above, during standby, the PIM **160**, **165** or **170** stores the local data on the client **110**, **115** or **120** in local.org. The user then depresses the "borrow me" button **545**. The security module **725** requests the user to enter a login and password, which the global server **105** authenticates. During these steps, it will be appreciated that the local data remains stored in local.org. Upon user identification and authentication, the global server **105** sends the workspace data **135** (global data) to the requesting client **110**, **115** or **120**. The instantiator **730** on the client **110**, **115** or **120** creates a new instance, e.g., a new window, of PIM API **740** and stores the received global data into another file, i.e., global.org. The data reader **745** and PIM API **740** enable the user to manipulate the global data, the manipulated data being referred to herein as "global data 2." Upon logout, the global data 2 is synchronized with the workspace data **135**. Namely, the general synchronization module **715** determines the changes made (Δglobal data), and synchronizes Δglobal data with the workspace data **135**. The de-instantiator **745** deletes global data 2, Δglobal data and global.org.

FIG. **8** is a flowchart illustrating a method **800** of accessing data remotely in accordance with the present invention. The method **800** begins with the processor **405** in step **805** opening the PIM **160**, **165** or **170** per user request, and the PIM **160**, **165** or **170** opening a PIM interface **500** (FIG. **5**). The PIM **160**, **165** or **170** in step **810** receives a "borrow me" request from the user, i.e., the user depresses the "borrow me" button **545**. The PIM API **740** in step **815** recognizes the request, and instructs the communications module **705** to create a communications link with the global server **105**.

The security module **725** in step **820** requests and transmits identification and authentication information such as login and password information from the user to the global server **105** for examination. If the global server **105** fails to identify or authenticate the user, then the method **800** ends. Otherwise, the instantiator **730** in step **825** opens a PIM interface **500** to display and enable manipulation of the workspace data **135** downloaded from the global server **105**. The data reader **735** in step **830** reads the workspace data **135** downloaded from the global server **105**, and in step **835** translates the data to the appropriate format if necessary. That is, the data reader **735** translates the workspace data **135** from the format implemented by the global server **105** to the format implemented by the PIM **160**, **165** or **170**. The PIM API **740** in step **840** passes the translated workspace data **135** to the PIM interfaces **500** and **600**.

The PIM **160**, **165** or **170** enables the user in step **845** to manipulate the workspace data **135** as necessary. Manipu-

lation includes adding new data, deleting workspace data **135**, editing workspace data **135**, etc. For example, the user can depress the e-mail button **515** in interface **500** to select, review and manipulate e-mail in interface **600**, and then can depress the calendar button **530** in interface **500** to select, review ad manipulate calendar information (not shown) in an interface similar to the e-mail interface **600**. In step **850**, the PIM API **740** waits to receive an "end session" request. Until an "end session" request is received, the method **800** returns to step **830** to enable continued data review and manipulation.

Upon receiving an "end session" or "unborrow me" request, the de-instantiator **745** initiates the general synchronization module **715** in step **855** to synchronize the manipulated workspace data on the client **110**, **115** or **120** with the workspace data **135** on the global server **105**, if required. Synchronization is described in greater detail with reference to FIG. **9**. The de-instantiator **745** in step **860** deletes the workspace data on the client **110**, **115** or **120**, and deletes all records of the matter. Method **800** then ends.

FIG. **9** is a flowchart illustrating a method **900** for synchronizing workspace data in a computer network **100**. Method **900** begins with the communications module **705** in step **905** establishing a communications link with the synchronization agent **130** of the global server **105**. The locator modules **710** in step **910** identify the memory locations of the workspace elements in the workspace data **135**. It will be appreciated that workspace element memory location identification may be implemented using intelligent software or dialogue boxes.

The general synchronization module **715** in step **915** compares version information (not shown) for each workspace element in the workspace data (on the client **110**, **115** or **120** and on the global server **105**) against a last synchronization signature to determine which workspace elements have been modified. In this embodiment, a workspace element may have been modified if the date and time of the last modification is after the date and time of the downloading.

If the general synchronization module **715** locates no modified workspace elements in the workspace data on the client **110**, **115** or **120**, then the method **900** ends. Otherwise, the general synchronization module in step **920** determines whether the version of the same workspace element of the workspace data **135** on the global server **105** has been modified since the data **135** was downloaded.

If only the version on the client **110**, **115** or **120** has been modified, then the general synchronization module **715** in step **925** stores the updated version of the workspace element at the global server **105**. To store the updated version on the global server **105**, the general synchronization module **715** may compute the changes made and forward the changes to the synchronization agent **130**. The synchronization agent **130** enters the changes into the global server **105** version. The general synchronization module **715** in step **930** determines whether all workspace elements downloaded to the client **110**, **115** or **120** have been examined. If not, then method **900** returns to step **915**. Otherwise, the synchronization agent **130** in step **935** updates the last synchronization signature, and method **900** ends. Updating the last synchronization signature will instruct the base systems **140** and **150** to synchronize the workspace data **145** and **155** with the workspace data **135** on the global server **105**, as described in the patent applications incorporated by reference above.

If the general synchronization module **715** in step **920** determines that both the version on the client **110**, **115** or **120** and the version on the global server **105** have been modified, then the general synchronization module in step **935** instructs the content-based synchronization module **729** to reconcile the modified versions. Reconciliation may include requesting instructions from the user, or performing based on pre-selected preferences responsive actions such as storing both versions at the global server **105**. The general synchronization module **715** in step **940** stores the preferred version on the global server **105**. Method **900** then proceeds to step **930**.

The foregoing description of the preferred embodiments of the present invention is by way of example only, and other variations and modifications of the above-described embodiments and methods are possible in light of the foregoing teaching. Although the network sites are being described as separate and distinct sites, one skilled in the art will recognize that these sites may be a part of an integral site, may each include portions of multiple sites, or may include combinations of single and multiple sites. Further, components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. Connections may be wired, wireless, modem, etc. The embodiments described herein are not intended to be exhaustive or limiting. The present invention is limited only by the following claims.

What is claimed is:

**1**. A computer-based method, comprising the steps of:

executing a workspace data manager on an untrusted client site;

requesting the workspace data manager to access data temporarily from a remote site, the remote being connected via a network to untrusted client site;

initiating a communications channel with the remote site;

downloading data from the remote site;

placing the data in temporary storage on the untrusted client site;

using the workspace data manager to present the downloaded data; and

automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

**2**. The method of claim **1**, further comprising the step of requesting the workspace data manager to provide an interface for enabling presentation of the downloaded data.

**3**. The method of claim **2**, further comprising the steps of using the workspace data manager to manipulate the dow data, thereby creating manipulated data, using the workspace data manager interface to request synchronization, and synchronizing the manipulated data with the data at the remote site.

**4**. The method of claim **3**, wherein the data at the remote site has not been modified after the step of downloading and before the step of synchronizing and therefore includes the downloaded data.

**5**. The method of claim **3**, wherein the data at the remote site has been modified after the step of downloading and before the step of synchronizing, and therefore is different than the downloaded data.

**6**. The method of claim **2**, wherein the workspace data manager provides an interface by creating an instance.

**7**. The method of claim **2**, wherein the workspace data manager provides an interface by providing access to its only interface.

**8**. The method of claim **1**, further comprising the step of translating the downloaded data from the format used by the remote site and the format used by the workspace data manager.

9. The method of claim 1, further comprising the step of deleting the workspace data manager interface after it is no longer needed.

10. A system on an untrusted client site, comprising:

a communications module for download data from a remote site, the remote site being connected via a network to the untrusted client site;

program code for placing the downloaded data in temporary storage on the untrusted client site;

an application program interface coupled to the communications module for communicating with a workspace data manager to present the downloaded data; and

program code coupled to the application program interface for automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

11. The system of claim 10, further comprising an instantiator for requesting the workspace data manager to provide an interface for enabling presentation of the downloaded data.

12. The system of claim 11, wherein the workspace manager enables manipulation of the downloaded data to create manipulated data and the data manipulation interface enables a request to synchronize the data, and further comprising a synchronization module coupled to the communications module for enabling synchronization of the manipulated data with the data at the remote site.

13. The system of claim 12, wherein the data stored at the remote site has not been modified and therefore includes the downloaded data.

14. The system of claim 12, wherein the data stored at the remote site has been modified, and therefore is different than the downloaded data.

15. The system of claim 14, further comprising a content-based synchronization module for synchronizing the data stored at the remote site with the manipulated data.

16. The system of claim 11, wherein the workspace data manager creates another instance of the interface to enable presentation of the downloaded data.

17. The system of claim 11, wherein the workspace data manager provides access to its only interface to enable presentation of the downloaded data.

18. The system of claim 11, further comprising a deinstantiator for deleting the interface after it is no longer required.

19. The system of claim 10, further comprising a data reader for translating the downloaded workspace data from the format used by the remote site to the format used by the workspace data manager.

20. A system comprising:

means for executing a workspace data manager on an untrusted client site;

means for requesting the workspace data manager to access data temporarily from a remote site, the remote site being connected via a network to the untrusted client site;

means for initiating a communications channel with the remote site;

means for downloading data from the remote site;

means for placing the data in storage on the untrusted client site;

means for using the workspace data manager to present the downloaded data; and

means for disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

21. A computer-readable storage medium storing program code for causing a computer to perform the steps of:

executing a workspace data manager on an untrusted client site;

requesting the workspace data manager to access data temporarily from a remote site, the remote site being connected via a network to the untrusted client site;

initiating a communications channel with the remote site;

downloading data from the remote site;

placing the data in temporary storage on the untrusted client site;

using the workspace data manager to present the downloaded data; and

automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

# CERTIFICATE OF CORRECTION

PATENT NO. : **6,151,606**

DATED : **November 21, 2000**

INVENTOR(S) : **Daniel J. Mendez**

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

**Column 12, line 47, after "manipulate the" change the word "dow" to -- downloaded --**
**Column 13, line 5, after "module for" change the word"download" to -- downloading --**
**Column 13, line 23, before "manager" insert -- data --**
**Column 14, line 18, after "placing the data in" insert -- temporary --**

Signed and Sealed this

Eighth Day of May, 2001

*Attest:*

NICHOLAS P. GODICI

*Attesting Officer*

*Acting Director of the United States Patent and Trademark Office*

# EXHIBIT B

US007039679B2

(12) **United States Patent**
**Mendez et al.**

(10) **Patent No.: US 7,039,679 B2**
(45) **Date of Patent: May 2, 2006**

(54) **SYSTEM AND METHOD FOR GLOBALLY AND SECURELY ACCESSING UNIFIED INFORMATION IN A COMPUTER NETWORK**

(75) Inventors: **Daniel J. Mendez**, Menlo Park, CA (US); **Mark D. Riggins**, Mercer Island, WA (US); **Prasad Wagle**, Santa Clara, CA (US); **Hong Q. Bui**, Cupertino, CA (US); **Mason Ng**, Moutain View, CA (US); **Sean Michael Quinlan**, San Fransico, CA (US); **Christine C. Ying**, Foster City, CA (US); **Christopher R. Zuleeg**, San Jose, CA (US); **David J. Cowan**, Menlo Park, CA (US); **Joanna A. Aptekar-Strober**, Menlo Park, CA (US); **R. Stanley Bailes**, San Jose, CA (US)

(73) Assignee: **Visto Corporation**, Redwood Shores, CA (US)

( * ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **10/741,113**

(22) Filed: **Dec. 19, 2003**

(65) **Prior Publication Data**

US 2004/0139178 A1 Jul. 15, 2004

**Related U.S. Application Data**

(63) Continuation of application No. 09/966,877, filed on Sep. 20, 2000, now Pat. No. 6,708,221, which is a continuation of application No. 08/903,118, filed on Jul. 30, 1997, now abandoned, and a continuation-in-part of application No. 08/865,075, filed on May 29, 1997, now Pat. No. 6,023,708, and a continuation-in-part of application No. 08/835,997, filed on Apr. 11, 1997, now Pat. No. 6,085,192, and a continuation-in-part of application No. 08/841,950, filed on Apr. 8, 1997, which is a continuation-in-part of application No. 08/766,307, filed on Dec. 13, 1996, now Pat. No. 6,131,116.

(51) **Int. Cl.**
*G06F 15/15* (2006.01)

(52) **U.S. Cl.** ..... **709/206**; 709/248
(58) **Field of Classification Search** ..... 709/206, 709/248, 202, 203, 100; 713/400; 370/350
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,652,698 A 3/1987 Hale et al.

(Continued)

OTHER PUBLICATIONS

US 5,373,559, 12/1994, Kaufman et al. (withdrawn)

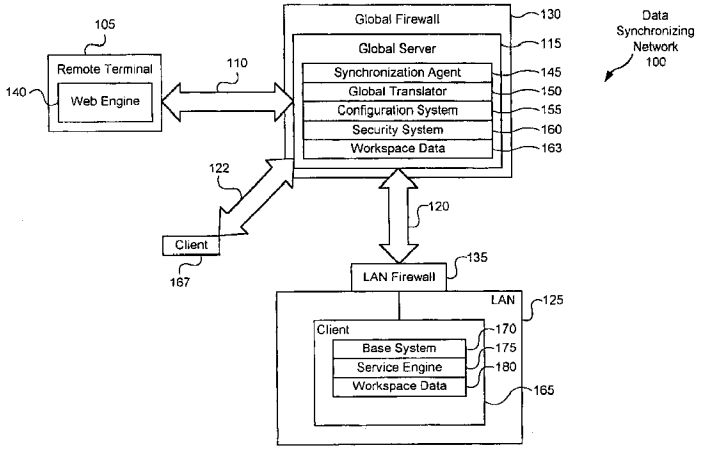
(Continued)

*Primary Examiner*—Mehmet B. Geckill
(74) *Attorney, Agent, or Firm*—Manatt Phelps & Phillips

(57) **ABSTRACT**

A client stores a first set of workspace data, and is coupled via a computer network to a global server. The client may be configured to synchronize portions of the first set of workspace data with the global server, which stores independently modifiable copies of the portions. The global server may also store workspace data which is not downloaded from the client, and thus stores a second set of workspace data. The global server may be configured to identify and authenticate a user seeking global server access from a remote terminal, and is configured to provide access to the first set or to the second set. Further, services may be stored anywhere in the computer network. The global server may be configured to provide the user with access to the services. The system may further include a synchronization-start module at the client site (which may be protected by a firewall) that initiates interconnection and synchronization with the global server when predetermined criteria have been satisfied.

**18 Claims, 15 Drawing Sheets**



U.S. PATENT DOCUMENTS


| | | | | |
|---|---|---|---|---|
| 4,714,995 | A | | 12/1987 | Materna et al. |
| 4,831,582 | A | | 5/1989 | Miller et al. |
| 4,875,159 | A | | 10/1989 | Cary et al. |
| 4,882,752 | A | | 11/1989 | Lindman et al. |
| 4,897,781 | A | | 1/1990 | Chang et al. |
| 4,916,738 | A | | 4/1990 | Chandra et al. |
| 5,048,085 | A | | 9/1991 | Abraham et al. |
| 5,150,407 | A | | 9/1992 | Chan |
| 5,220,603 | A | | 6/1993 | Parker |
| 5,263,157 | A | | 11/1993 | Janis |
| 5,265,159 | A | | 11/1993 | Kung |
| 5,333,266 | A | | 7/1994 | Boaz et al. |
| 5,386,564 | A | | 1/1995 | Shearer et al. |
| 5,388,255 | A | | 2/1995 | Pytlik et al. |
| 5,392,390 | A | | 2/1995 | Crozier |
| 5,420,927 | A | | 5/1995 | Micali |
| 5,425,102 | A | | 6/1995 | Moy |
| 5,434,918 | A | | 7/1995 | Kung et al. |
| 5,483,596 | A | | 1/1996 | Rosenow et al. |
| 5,491,752 | A | | 2/1996 | Kaufman et al. |
| 5,495,533 | A | | 2/1996 | Linehan et al. |
| 5,510,777 | A | | 4/1996 | Pilc et al. |
| 5,544,320 | A | | 8/1996 | Konrad |
| 5,544,322 | A | | 8/1996 | Cheng et al. |
| 5,572,643 | A | | 11/1996 | Judson |
| 5,581,749 | A | | 12/1996 | Hossain et al. |
| 5,588,132 | A | | 12/1996 | Cardoza |
| 5,600,834 | A | | 2/1997 | Howard |
| 5,604,788 | A | | 2/1997 | Tett |
| 5,613,012 | A | | 3/1997 | Hoffman et al. |
| 5,623,601 | A | | 4/1997 | Vu |
| 5,627,658 | A | | 5/1997 | Connors et al. |
| 5,627,997 | A | | 5/1997 | Pearson et al. |
| 5,632,011 | A | | 5/1997 | Landfield et al. |
| 5,634,053 | A | | 5/1997 | Noble et al. |
| 5,644,354 | A | | 7/1997 | Thompson et al. |
| 5,647,002 | A | | 7/1997 | Brunson |
| 5,652,884 | A | | 7/1997 | Palevich |
| 5,657,390 | A | | 8/1997 | Elgamal et al. |
| 5,664,207 | A | | 9/1997 | Crumpler et al. |
| 5,666,530 | A | | 9/1997 | Clark et al. |
| 5,666,553 | A | | 9/1997 | Crozier |
| 5,675,782 | A | | 10/1997 | Montague et al. |
| 5,678,039 | A | | 10/1997 | Hinks et al. |
| 5,680,542 | A | | 10/1997 | Mulchandani et al. |
| 5,682,478 | A | | 10/1997 | Watson et al. |
| 5,682,524 | A | | 10/1997 | Freund et al. |
| 5,684,951 | A | | 11/1997 | Goldman et al. |
| 5,684,984 | A | | 11/1997 | Jones et al. |
| 5,684,990 | A | | 11/1997 | Boothby |
| 5,687,322 | A | | 11/1997 | Deaton et al. |
| 5,701,400 | A | | 12/1997 | Amado |
| 5,701,423 | A | | 12/1997 | Crozier |
| 5,706,427 | A | | 1/1998 | Tabuki |
| 5,706,502 | A | | 1/1998 | Foley et al. |
| 5,710,918 | A | | 1/1998 | Lagarde et al. |
| 5,710,922 | A | | 1/1998 | Alley et al. |
| 5,713,019 | A | | 1/1998 | Keaten |
| 5,715,403 | A | | 2/1998 | Stefik |
| 5,717,925 | A | | 2/1998 | Harper et al. |
| 5,721,779 | A | | 2/1998 | Funk |
| 5,721,908 | A | | 2/1998 | Lagarde et al. |
| 5,721,914 | A | | 2/1998 | DeVries |
| 5,727,202 | A | | 3/1998 | Kucala |
| 5,729,735 | A | | 3/1998 | Meyering |
| 5,742,668 | A | | 4/1998 | Pepe et al. |
| 5,745,360 | A | | 4/1998 | Leone et al. |
| 5,752,059 | A | | 5/1998 | Holleran et al. |
| 5,752,246 | A | | 5/1998 | Rogers et al. |
| 5,754,830 | A | | 5/1998 | Butts et al. |
| 5,757,916 | A | | 5/1998 | MacDoran et al. |
| 5,758,150 | A | | 5/1998 | Bell et al. |
| 5,758,354 | A | | 5/1998 | Huang et al. |
| 5,758,355 | A | | 5/1998 | Buchanan |
| 5,764,902 | A | | 6/1998 | Rothrock |
| 5,765,171 | A | | 6/1998 | Gehani et al. |
| 5,768,510 | A | | 6/1998 | Gish |
| 5,778,346 | A | | 7/1998 | Frid-Nielsen et al. |
| 5,784,463 | A | | 7/1998 | Chen et al. |
| 5,784,464 | A | | 7/1998 | Akiyama et al. |
| 5,787,172 | A | | 7/1998 | Arnold |
| 5,790,425 | A | | 8/1998 | Wagle |
| 5,790,790 | A | | 8/1998 | Smith et al. |
| 5,790,974 | A | | 8/1998 | Tognazzini |
| 5,794,252 | A | | 8/1998 | Bailey et al. |
| 5,799,086 | A | | 8/1998 | Sudia |
| 5,799,318 | A | | 8/1998 | Cardinal et al. |
| 5,802,530 | A | | 9/1998 | Van Hoff |
| 5,812,398 | A | | 9/1998 | Nielsen |
| 5,812,668 | A | | 9/1998 | Weber |
| 5,812,773 | A | | 9/1998 | Norin |
| 5,815,683 | A | | 9/1998 | Vogler |
| 5,818,935 | A | | 10/1998 | Maa |
| 5,828,840 | A | | 10/1998 | Cowan et al. |
| 5,832,483 | A | | 11/1998 | Barker |
| 5,835,087 | A | | 11/1998 | Herz et al. |
| 5,835,601 | A | | 11/1998 | Shimbo et al. |
| 5,845,282 | A | | 12/1998 | Alley et al. |
| 5,857,201 | A | | 1/1999 | Wright, Jr. et al. |
| 5,862,325 | A | | 1/1999 | Reed et al. |
| 5,862,346 | A | | 1/1999 | Kley et al. |
| 5,870,544 | A | | 2/1999 | Curtis |
| 5,870,759 | A | | 2/1999 | Bauer et al. |
| 5,870,765 | A | | 2/1999 | Bauer et al. |
| 5,878,230 | A | | 3/1999 | Weber et al. |
| 5,909,689 | A | | 6/1999 | Van Ryzin |
| 5,924,103 | A | | 7/1999 | Ahmed et al. |
| 5,928,329 | A | | 7/1999 | Clark et al. |
| 5,943,676 | A | | 8/1999 | Boothby |
| 5,951,652 | A | | 9/1999 | Ingrassia, Jr. et al. |
| 5,961,590 | A | | 10/1999 | Mendez et al. |
| 5,966,714 | A | | 10/1999 | Huang et al. |
| 5,968,131 | A | | 10/1999 | Mendez et al. |
| 5,974,238 | A | | 10/1999 | Chase, Jr. |
| 5,982,898 | A | | 11/1999 | Hsu et al. |
| 5,987,609 | A | | 11/1999 | Hasebe |
| 5,999,932 | A | | 12/1999 | Paul |
| 5,999,947 | A | | 12/1999 | Zollinger et al. |
| 6,006,017 | A | | 12/1999 | Joshi et al. |
| 6,006,274 | A | * | 12/1999 | Hawkins et al. ..... 709/248 |
| 6,020,885 | A | | 2/2000 | Honda |
| 6,021,427 | A | | 2/2000 | Spagna et al. |
| 6,023,700 | A | | 2/2000 | Owens et al. |
| 6,023,708 | A | | 2/2000 | Mendez et al. |
| 6,034,621 | A | | 3/2000 | Kaufman |
| 6,052,735 | A | | 4/2000 | Ulrich et al. |
| 6,073,165 | A | | 6/2000 | Narasimhan et al. |
| 6,085,192 | A | | 7/2000 | Mendez et al. |
| 6,094,477 | A | | 7/2000 | Nada et al. |
| 6,108,691 | A | | 8/2000 | Lee et al. |
| 6,108,709 | A | | 8/2000 | Shinomura et al. |
| 6,118,856 | A | | 9/2000 | Paarsmarkt et al. |
| 6,125,281 | A | | 9/2000 | Wells et al. |
| 6,131,096 | A | | 10/2000 | Ng et al. |
| 6,131,116 | A | | 10/2000 | Riggins et al. |
| 6,138,146 | A | | 10/2000 | Moon et al. |
| 6,151,606 | A | | 11/2000 | Mendez |
| 6,154,844 | A | | 11/2000 | Touboul et al. |
| 6,169,986 | B1 | | 1/2001 | Bowman et al. |
| 6,182,118 | B1 | | 1/2001 | Finney et al. |
| 6,212,529 | B1 | * | 4/2001 | Boothby et al. ..... 707/201 |

| | | | | |
|---|---|---|---|---|
| 6,249,805 | B1 | | 6/2001 | Fleming, III |
| 6,295,541 | B1 | | 9/2001 | Bodnar et al. |
| 6,304,881 | B1 | | 10/2001 | Halim et al. |
| 6,311,186 | B1 | | 10/2001 | MeLampy et al. |
| 6,317,797 | B1 | * | 11/2001 | Clark et al. ..... 710/5 |
| 6,324,542 | B1 | | 11/2001 | Wright, Jr. et al. |
| 6,334,140 | B1 | | 12/2001 | Kawamata |
| 6,343,313 | B1 | | 1/2002 | Salesky et al. |
| 6,389,455 | B1 | | 5/2002 | Fuisz |
| 6,438,583 | B1 | | 8/2002 | McDowell et al. |
| 6,446,090 | B1 | | 9/2002 | Hart |
| 6,477,545 | B1 | | 11/2002 | LaRue |
| 6,510,455 | B1 | | 1/2003 | Chen et al. |
| 6,564,218 | B1 | | 5/2003 | Roth |
| 6,631,416 | B1 | | 10/2003 | Bendinelli et al. |
| 6,697,942 | B1 | | 2/2004 | L'Heureux et al. |

OTHER PUBLICATIONS

IntelliLink Corporation, IntelliLink for Windows Release 3.0, "User's Guide" 1994, Nashua, NH.

Lotus Development Corporation, Lotus Notes Release 4,"Application Developer's Guide" 1995, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 3.3 North American Server Edition, "Lotus Notes, the Groupware Standard" 1994, Cambridge, MA.

Sams Publishing Dahl, Andrew, "Lotus Notes 4 Administrator's Survival Guide," 1996, Indianapolis, IN.

Advisor Publications –Lotus Notes Advisor, Pyle, Hugh, "The Notes Architecture," 1995.

Advisor Publications –Lotus Notes Advisor, Augun, Audry, "Integrating Lotus Notes with Enterprise Data," 1996.

Advisor Publicatons –Lotus Notes Advisor, Opyt, Barbara and Dale, Robert, "Use the Internet as Your Lotus Notes WAN," 1996.

Lotus Development Corporation, Lotus Notes Knowledge Base, "What is the Notes Replicator?" 1995, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Knowledge Base, "Firewall Security Overview and How Firewalls Relate to Lotus Notes" 1996, Cambridge, MA.

Network Computing, Frenkel, Garry, "Pumping for Info: Notes and Database Integration," 1996.

IBM Corporation, Hawker et al., "Secrets to Running Lotus Notes: The Decisions No One Tells You How to Make," 1996, Research Triangle Park, NC.

Lotus Development Corporation, InterNotes Web Publisher Release 4, "InterNotes Wed Publisher Guide" 1996, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 4,"Database Manager's Guide" 1995, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 4,"Administrator's Guide" 1995, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 4, "Deployment Guide" 1995, Cambridge, MA.

IBM Lotus Technical Library, Lotus Documentation, "Lotus Notes Internet Cookbook for Notes Release 3", Jan. 16, 1996, pp. 1–26 http://www–12.lotus.com.ldd/doc/domino notes/cookbook/cbookv4.nsf/e 12503289bf7b3a385256.

John Wiley & Sons –Wiley Computer Publishing, Falkner, Mike, "How to Plan, Develop, and Implement Lotus Notes in Your Organization " 1996, USA4.

McGraw–Hill, Lamb, LJohn P and Lew, Peter W., "Lotus Notes Network Design for Notes Release 3 and 4, " 1996, Quebecor–Fairfield, PA.

Lotus Notes, "Overview –What is Lotus NotesPump?", including "Notes Pum 1.0 Release Notes".

Lotus Development Corporation, Lotus Notes Release 3.1, The groupware standard, "Site and System Planning Guide" 1994, Cambridge MA.

Lotus Development Corporation, Lotus Notes Release 3.1, The groupware standard, "Site and System Planning Guide" 1994, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 3.1. The groupwarestandard, "Administrator's Guide Server for NetWare, OS/2, and UNIX" 1994, Cambridge, MA.

IBM Lotus Technical Library, Lotus Documentation,"Lotus Notes Internet Cookbook for Notes Release 4", Date: Feb. 14, 1996, pp. 1–30. *http://www–12.lotus.com/ldd/doc/domino notes/cookbook/cbookv4.nsf/e12503289bf7b3a385256.*

Kistler, James J. and Satyanarayanan, M., "Disconnected Operation in the Coda File System," ACM Transactions on Computer Systems, vol. 10, No. 1, Feb. 1992, pp. 3–25.

Hills, Alex, and Johnson, David B., "Wireless Data Network Infrastructure at Carnegie Mellon University," IEEE Personal Communications, 3(1), Feb. 1996.

Satyanarayanan, Mahadev, "Mobile Information Access," IEEE Personal Communications, Feb. 1996, pp. 26–33.

Satyanarayanan, Mahadev et al., "Coda: A Highly Available File System for a Distributed Workstation Environment," IEEE Transactions on Computers, vol. 39, No. 4, Apr. 1990, pp. 447–59.

Satyanarayanan, Mahadev, "Scalable, Secure, and Highly Available Distributed File Access," Computer, May 1990, pp. 9–21.

Mummert, Lily B. et al., "Exploiting Weak Connectivity for Mobile File Access," SIGOPS '95, dEC. 1995, PP. 143–55.

Terry, Douglas B. et al., "Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System," SIGOPS '95, dEC. 1995, PP. 172–183.

Demers, Alan et al., "The Bayou Architecture: Support for Data Sharing among Mobile Users," Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, Dec. 1994, pp. 2–7.

Petersen, Karin et al., "Bayou: Replicated Database Services for World–wide Applictions," Proceedings Seventh ACM SIGOPS European Workshop (EuroSIIGOPS'96), Connemara, Ireland, 1996, pp. 275–280.

Crocker, David H., "RFC822: Standard for ARPA Internet Text Messages," http://www.w3.org/Protocols/rfc822/.

Theisen, Tim, "AFS distributed filesystem FAQ (1/2)," posted to uwisc.general newsgroup, Jul. 25, 1994.

Theisen, Tim, "AFS distributed filesystem FAQ (2/2)," posted to uwisc.general newsgroup, Jul. 25, 1994.

Schubert, Eric, "Re: telnet/internet and ... ," posted to comp.sys.hp.mpe newsgroup, Jun. 20, 1995.

Glazman, Daniel, "SOFTWARE: HHTPtool [sic]1.1, a file transfer utility over HTTP using PUT and GET," posted to comp.infosystems.www.announce newsgroup, May 17, 1995.

Glazman, Daniel, "SOFTWARE:HTTPtool v1.0 for Windows3.x, file transfer utility over HTTP," posted to comp.infosystems.www.announce newsgroup, Mar. 21, 1996.

Singhal, M., "Update transport: A new technique for update synchronization in replicated database systems," IEEE Transactions on Software Engineering, vol. 16, No. 12, Dec. 1990, pp. 1325–1336.

Rao, H. & Skarra, A., "A transport service for synchronized replication across loosely–connected file systems," IEEE Transactions on Software Engineering, Apr. 1995, pp. 110–117.
Crispin, M., "Internet Message Access Protocol –RFC 1730 Version 4," Dec. 1994, pp. 1–52, http://www.faqs.org/rfcs/rfc1730.html.
Rao, Venkat & Aline, Mary, "Burrowing through firewalls," Dec. 1996, pp. 1–5, http://java.sun.com/developer/technicalAreticles/InnerWorkings/Burrowing/.
Mason, Justin, "Tunneling over HTTP," Dec. 11, 1996, pp. 1–2 http://www.netsys.com/firewalls–9612/0488.html.
Elgamal, Taher, "The Secure Sockets Layer Protocol (SSL)," agenda for the Danvers IETF meeting, Apr. 1995, pp. 1–5 –http://ww.ietf.cnri.reston.va.us/proceedings/95apr/sec/cat.elgamal.slides.html.
Gray, Terry, "Message Access Paradigms and Protocols," revised Sep. 28, 1995, pp. 1–11 http://www.imap.org/imap.vs.pop.html.
Ouellette, Tim, "Data for everyone, bills for none?," Computerworld, Mar. 17, 1997, pp. 43, 46.
Angus, Jeffrey G., "Sales force automation has GoldMine," Computerworld, Oct. 7, 1996, p. 59.
Radosevich, Linda, "Users want unified mail directories," Computerworld, Aug. 30, 1993, p. 12.
Bruno, Charles, "Firm pushes E–mail limits beyond rivals', " Network World, Aug. 12, 1991, pp. 33, 53.
Sliwa, Carol and Cole, Barb, "MESA declares a groupware dente," Network World, Aug. 5, 1996, p. 29.
Grosse, Eric, "Repository Mirroring," ACM Transactions on Mathematical Software, vol. 21, No. 1, Mar. 199, pp. 89–97.
Schilit, Bill N. and Theimer, Marvin M., "Disseminating Active Map Information to Mobile Hosts," IEEE Network, Sep./Oct. 1994, pp. 22–32.
Levy, Eliezer and Silberschatz, Abraham, "Distributed File Systems: Concepts and Examples," ACM Computing Surveys, vol. 22, No. 4, Dec. 1990, pp. 321–74.
Rymer, John R., "The Muddle in the Middle," Byte, Apr. 1996, pp. 67–70.
Baum, David, "Intranet Politics and Technologies," Byte, May 1997, pp. 88A–88H.
Udell, John "Push Me, Pull You," Byte, Sep. 1996, pp. 117–120.
Kador, John, "The Ultimate Middleware," Byte, Apr. 1996, pp. 79–83.
Salamone, Salvatore, "Middle(ware) Management," Byte, Apr. 1996, pp. 71–76.
Nance, Barry, "Balance the Load with Transaction Server," Byte, Jun. 1997, pp. 81–84.
Francett, Barbara, "Replication on the Run," Software Magazine, Aug. 1996, pp. 63–66.
Darling, Charles B., EDA/SQL Loses a Little and Gains a Lot, Datamation, May 1, 1996, p. 12.
Varney, Sarah E., "Arm your Salesforce with the Web," Datamation Oct. 1996, pp. 72–74.
Fulcher, Jim, "Is it or isn't it?, " Manufacturing Systems, Oct. 1996, pp. 56–61.
Lamb, John and Cusato, Tony, "LAN–Based Office for the Enterprise, A Case Study," Proceedings, 19th Conference on Local Computer Networks, Minneapolis, Minnesota, Oct. 2–5, 1994, pp. 440–447.
Kawell Jr., Leonard et al., "Replicated Document Management In A Group Communications System," presented at the Second Conference on Computer–Supported Cooperative Work, Portland, Oregon, Sep. 26–28, 1988, as printed in Groupware: Software for Computer–Supported Cooperative Work, IEEE Computer Society Press, pp. 226–235.
Hong, Jack et al., "Personal Electronic Notebook with Sharing,", Proceedings of the Fourth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Berkeley–Springs, West Virginia, Apr. 20–22, 1995, pp. 88–94.
Mace, Scott, "DataSync 2.0 enhances synchronization of data," InfoWorld, Jun. 6, 1994, p. 28.
Parker, D. Stott Jr. et al., "Detection of Multual Inconsistency in Distributed Systems," IEEE Transactions on Software Engineering, vol. SE–9, No. 3, May. 1983, pp. 240–246.
Satayanarayanan, Mahadev et al., "Coda: A Highly Available File System for a Distributed Workstation Environment," IEEE Transactions on Computers, vol. 39, No. 4, Apr. 1990, pp. 447–458.
Ceri, Stefano et al., "The Case for Independent Updates," Second Workshop on the Management of Replicated Data, Monterey, California, Nov. 12–13, 1992, pp. 17–19.
Downing, Alan R. et al., "OSCAR: A System for Weak–Consistency Replication," Proceedings, Workshop on the Management of Replicated Data, Houston, Texas, Nov. 8–9, 1990, pp. 26–30.
Siegel, Alex et al., "Deceit: A Flexible Distributed File Systems," Proceedings of Summer 1990 USENIX Conference, Anaheim, California, Jun. 11–15, 1990, pp. 51–61.
Chutani, Sailesh, "The Episode File System," Conference Proceedings, USENIX Winter 1992, Technical Conference, San Francisco, California, Jan. 20–24, 1992, pp. 43–59.
Seltzer, Margo, "An Implementation of Log–Structured File System for UNIX," Conference Proceedings, USENIX Winter 1993 Technical Conference, San Diego, California, Jan. 25–29, 1993, pp. 307–326.
Vahalia, Uresh, "Metadata Logging in an NFS Server," Conference Proceedings USENIX 1995 Technical Conference on UNIX and Advanced Computing Systems, New Olreans, Louisiana, Jan. 16–20, 1995, pp. 265–276.

* cited by examiner

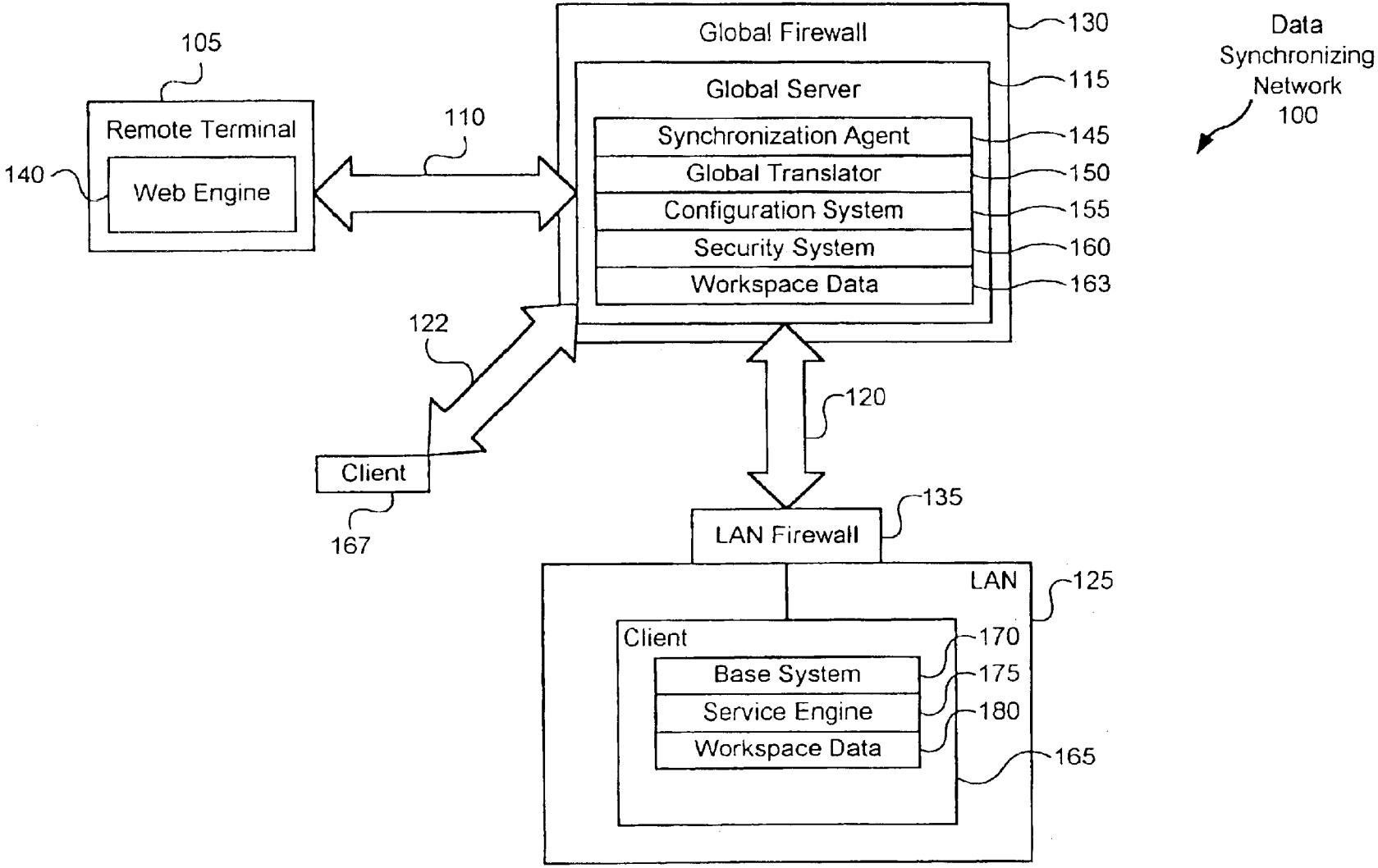

Data Synchronizing Network
100
105
Remote Terminal
140
Web Engine
110
Global Firewall
130
Global Server
115
Synchronization Agent
145
Global Translator
150
Configuration System
155
Security System
160
Workspace Data
163
122
120
Client
167
LAN Firewall
135
LAN
125
Client
170
Base System
175
Service Engine
180
Workspace Data
165

FIG. 1

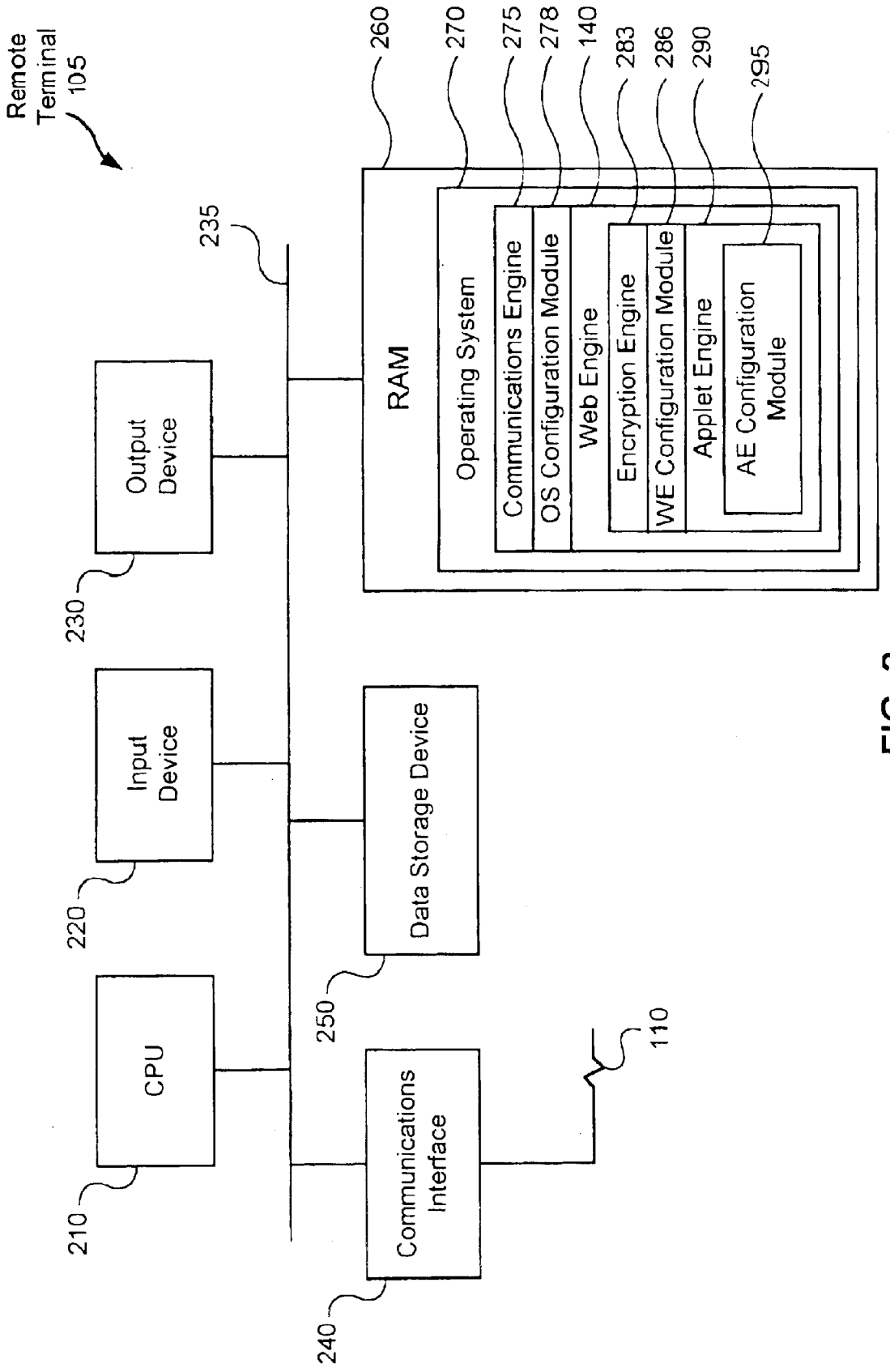

Remote Terminal 105
235
RAM
260
Operating System
270
Communications Engine
275
OS Configuration Module
278
Web Engine
140
Encryption Engine
283
WE Configuration Module
286
Applet Engine
290
AE Configuration Module
295
Output Device
230
Input Device
220
CPU
210
Data Storage Device
250
Communications Interface
240
110

FIG. 2

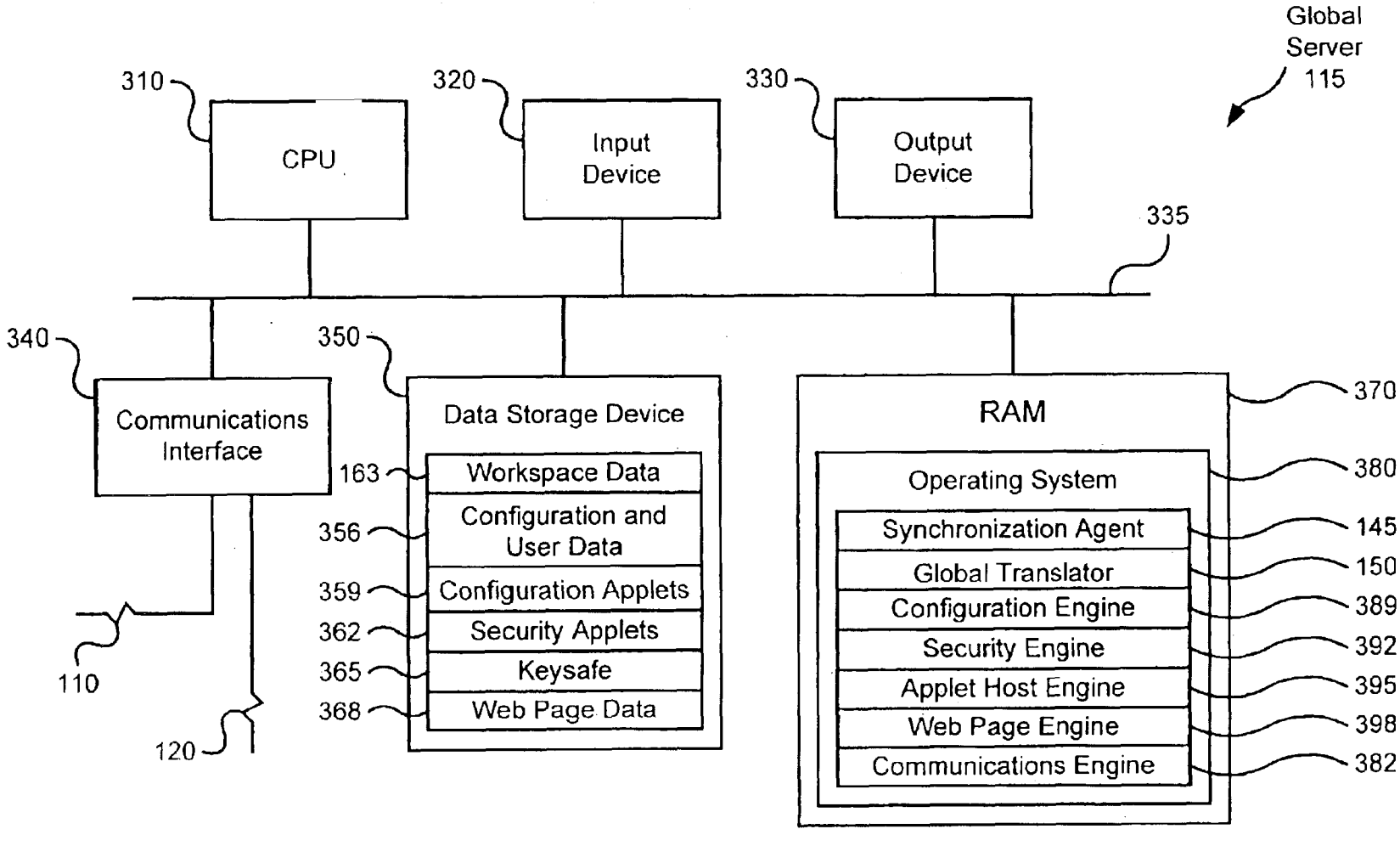

Global Server 115
310
CPU
320
Input Device
330
Output Device
335
340
Communications Interface
350
Data Storage Device
163
Workspace Data
356
Configuration and User Data
359
Configuration Applets
362
Security Applets
365
Keysafe
368
Web Page Data
110
120
RAM
370
Operating System
380
Synchronization Agent
145
Global Translator
150
Configuration Engine
389
Security Engine
392
Applet Host Engine
395
Web Page Engine
398
Communications Engine
382

FIG. 3

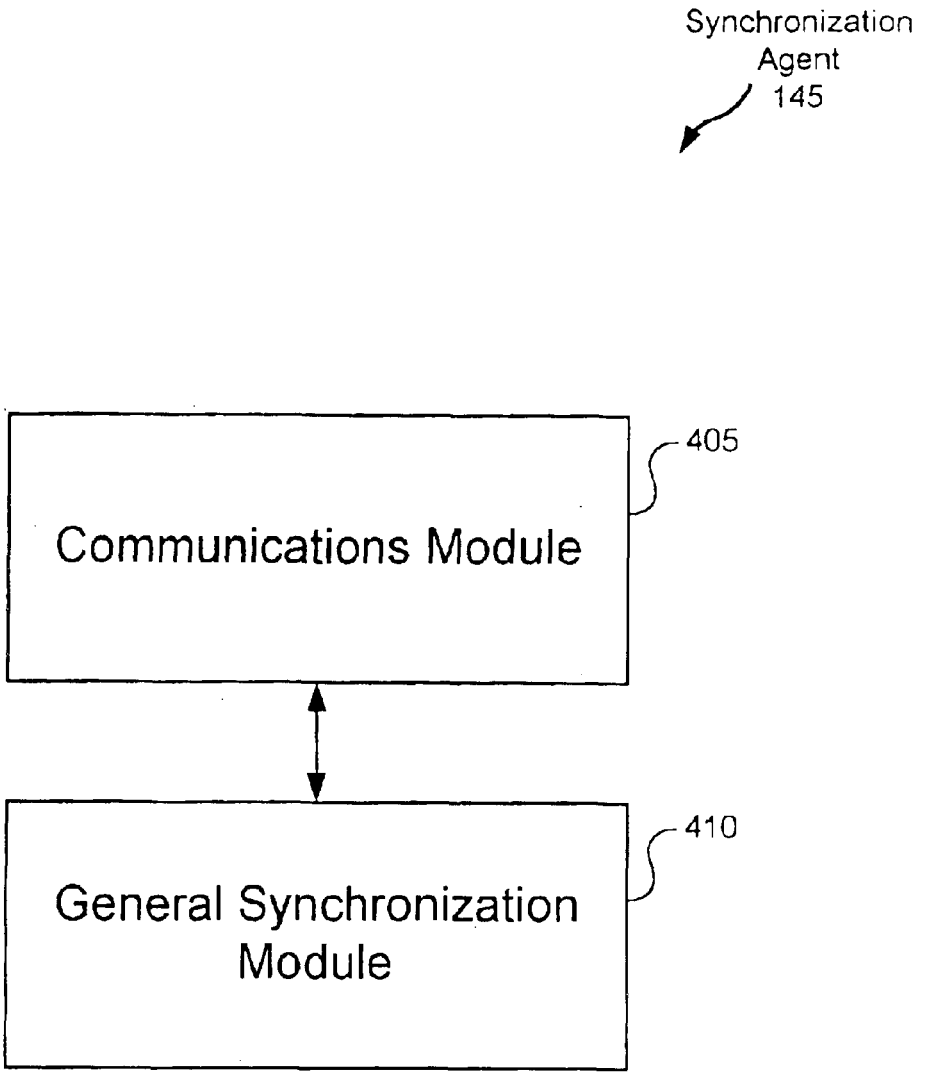

Synchronization Agent 145
Communications Module
405
General Synchronization Module
410

FIG. 4

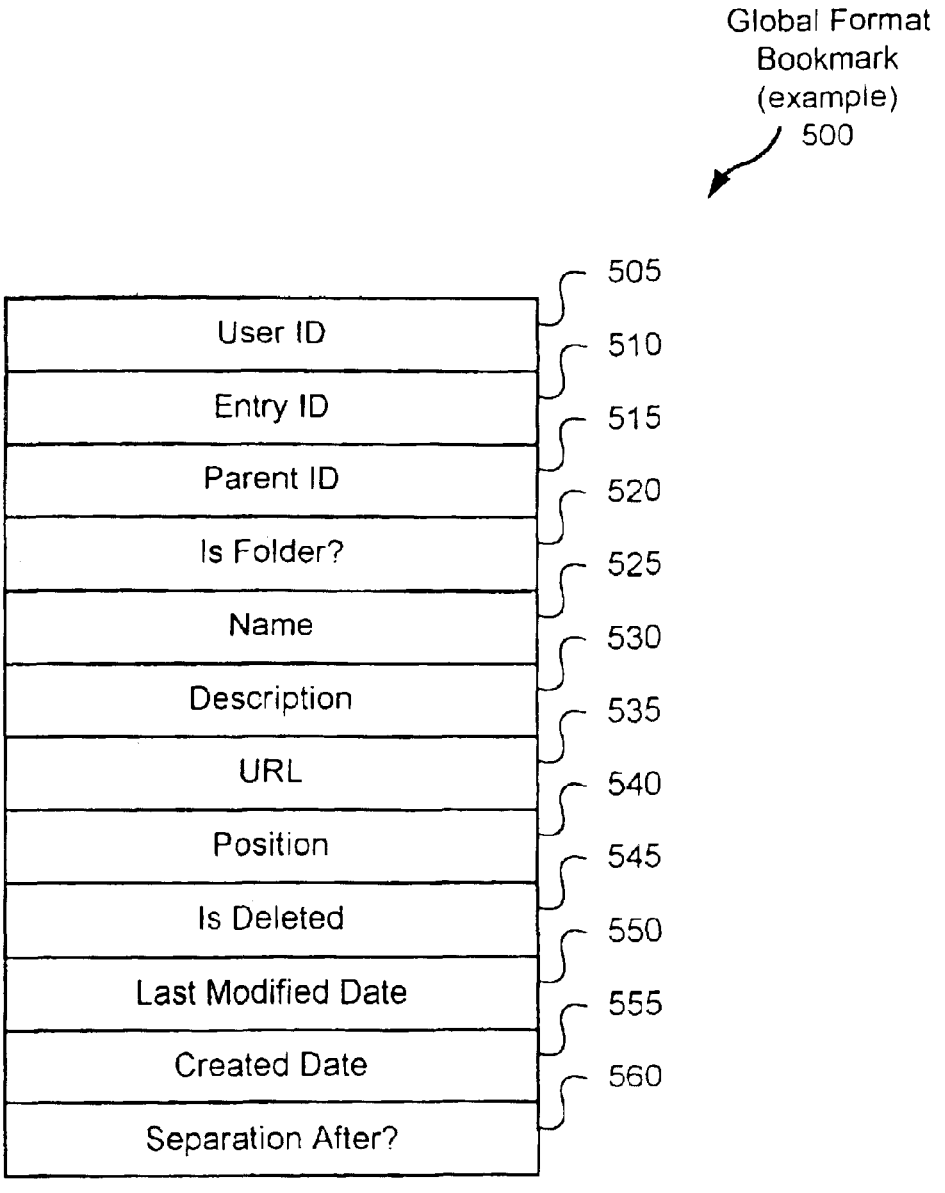

Global Format Bookmark (example)
500
User ID
505
Entry ID
510
Parent ID
515
Is Folder?
520
Name
525
Description
530
URL
535
Position
540
Is Deleted
545
Last Modified Date
550
Created Date
555
Separation After?
560

FIG. 5

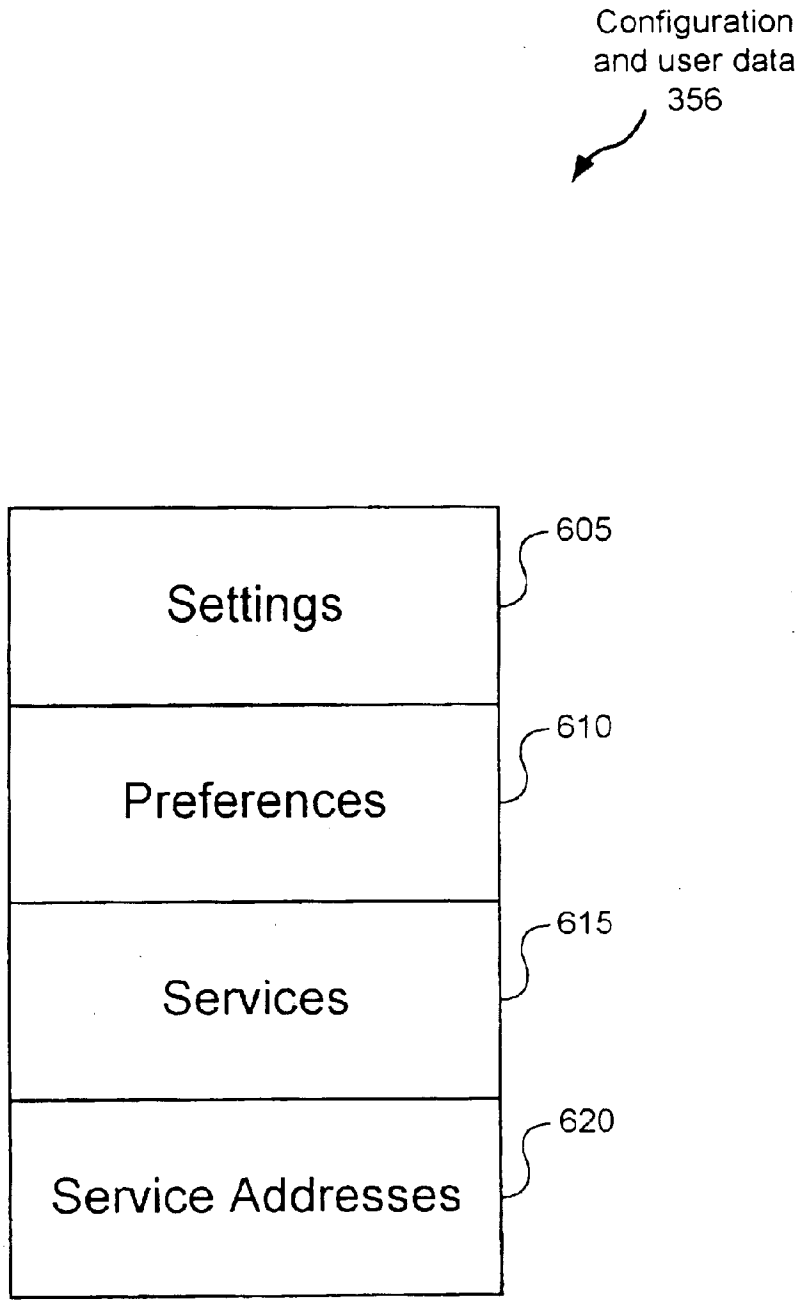

Configuration and user data 356
Settings
605
Preferences
610
Services
615
Service Addresses
620

FIG. 6

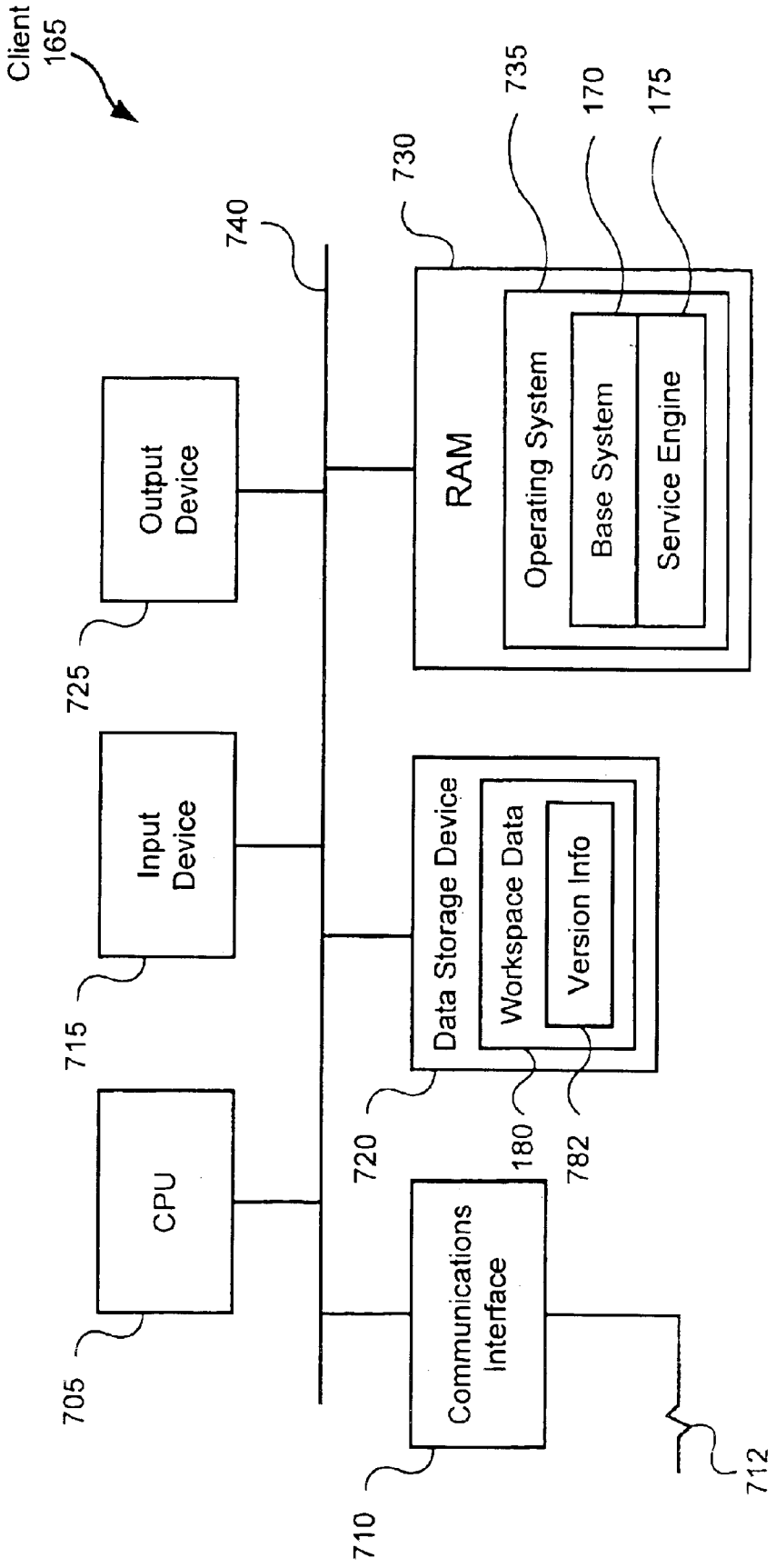

Client
165
740
730
735
170
175
RAM
Operating System
Base System
Service Engine
Output Device
725
Input Device
715
Data Storage Device
Workspace Data
Version Info
720
180
782
CPU
705
Communications Interface
710
712

FIG. 7

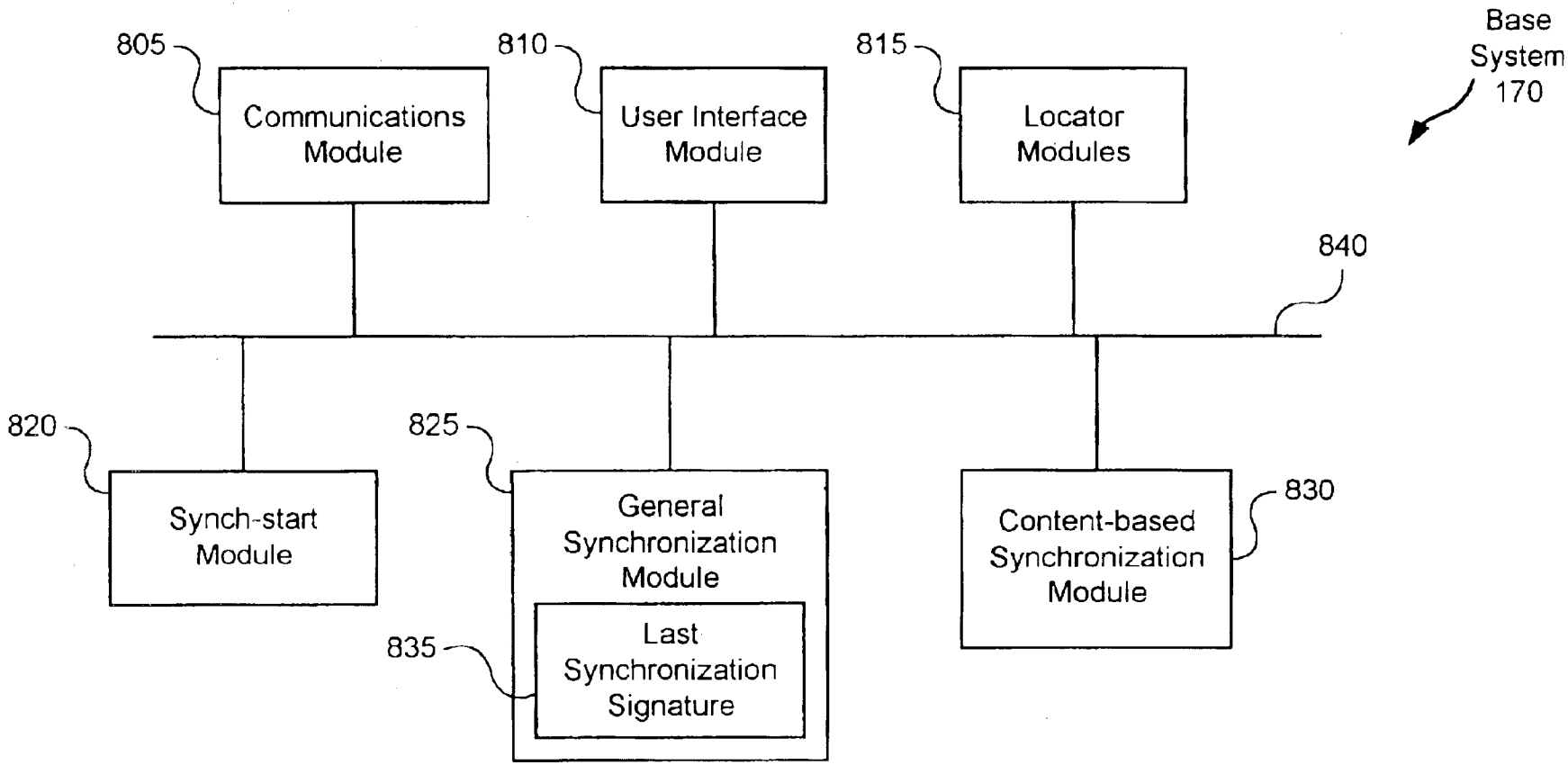

805
Communications Module
810
User Interface Module
815
Locator Modules
Base System 170
840
820
Synch-start Module
825
General Synchronization Module
835
Last Synchronization Signature
830
Content-based Synchronization Module

FIG. 8

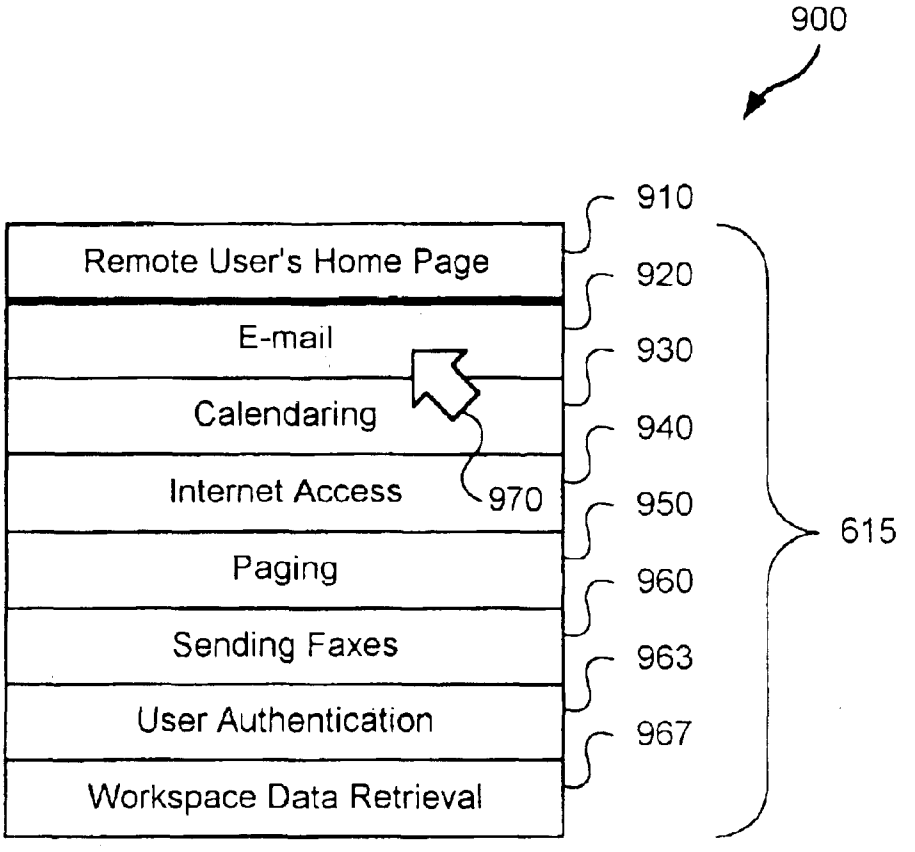

900
910
Remote User's Home Page
920
E-mail
930
Calendaring
940
Internet Access
970
950
615
Paging
960
Sending Faxes
963
User Authentication
967
Workspace Data Retrieval

FIG. 9

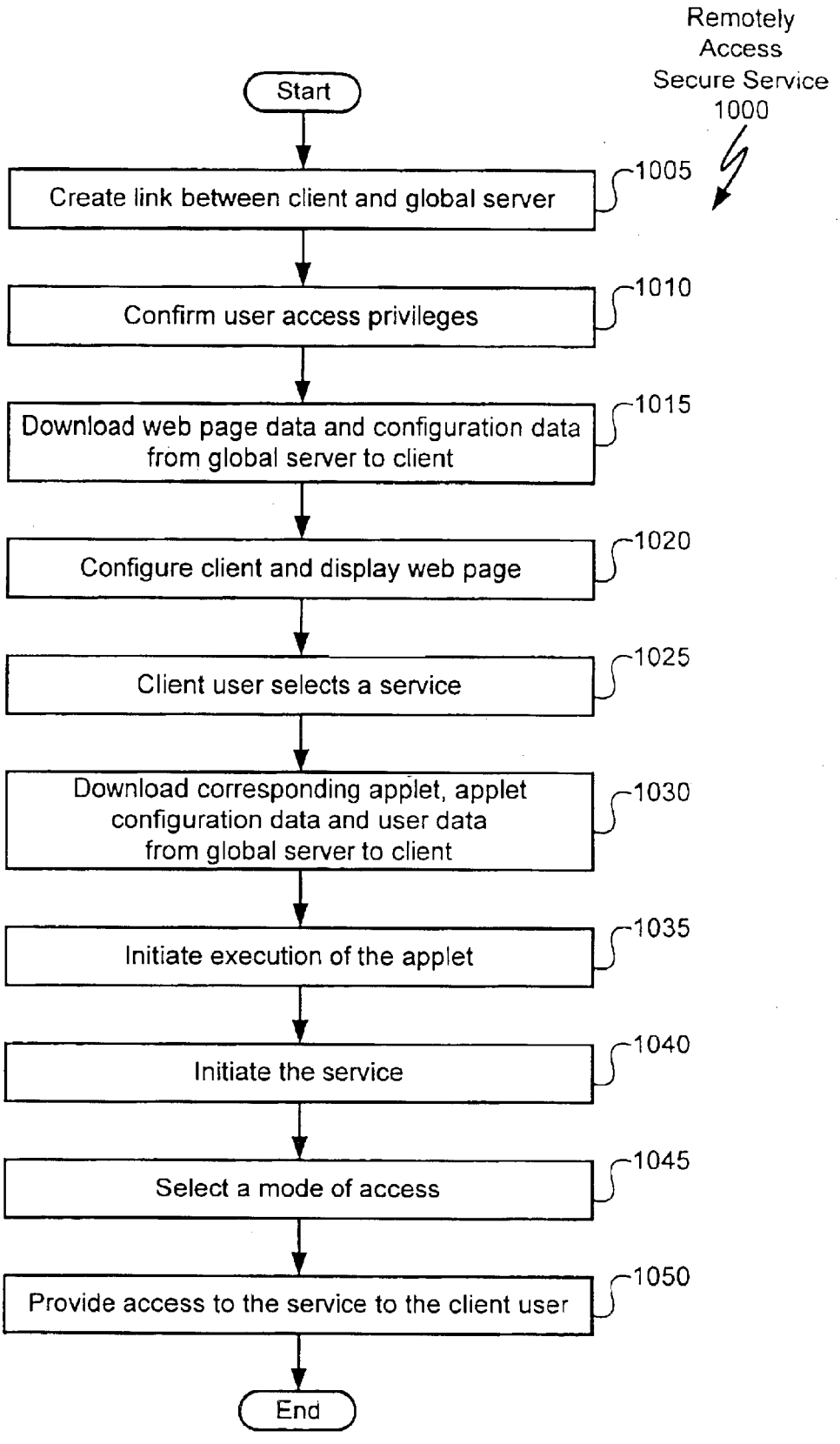

Remotely Access Secure Service 1000
Start
Create link between client and global server
1005
Confirm user access privileges
1010
Download web page data and configuration data from global server to client
1015
Configure client and display web page
1020
Client user selects a service
1025
Download corresponding applet, applet configuration data and user data from global server to client
1030
Initiate execution of the applet
1035
Initiate the service
1040
Select a mode of access
1045
Provide access to the service to the client user
1050
End

FIG. 10

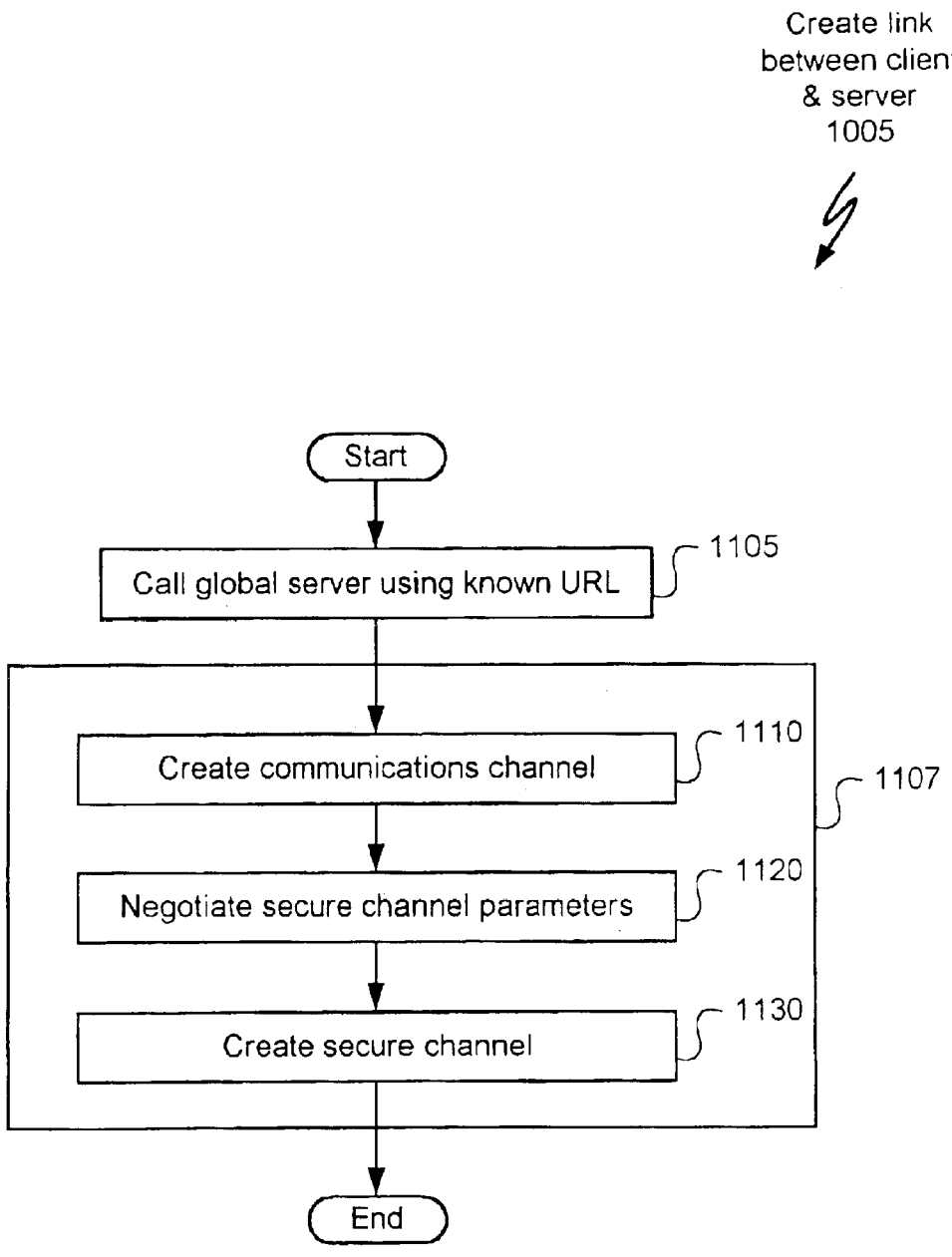

Create link between client & server 1005
Start
Call global server using known URL
1105
Create communications channel
1110
1107
Negotiate secure channel parameters
1120
Create secure channel
1130
End

FIG. 11

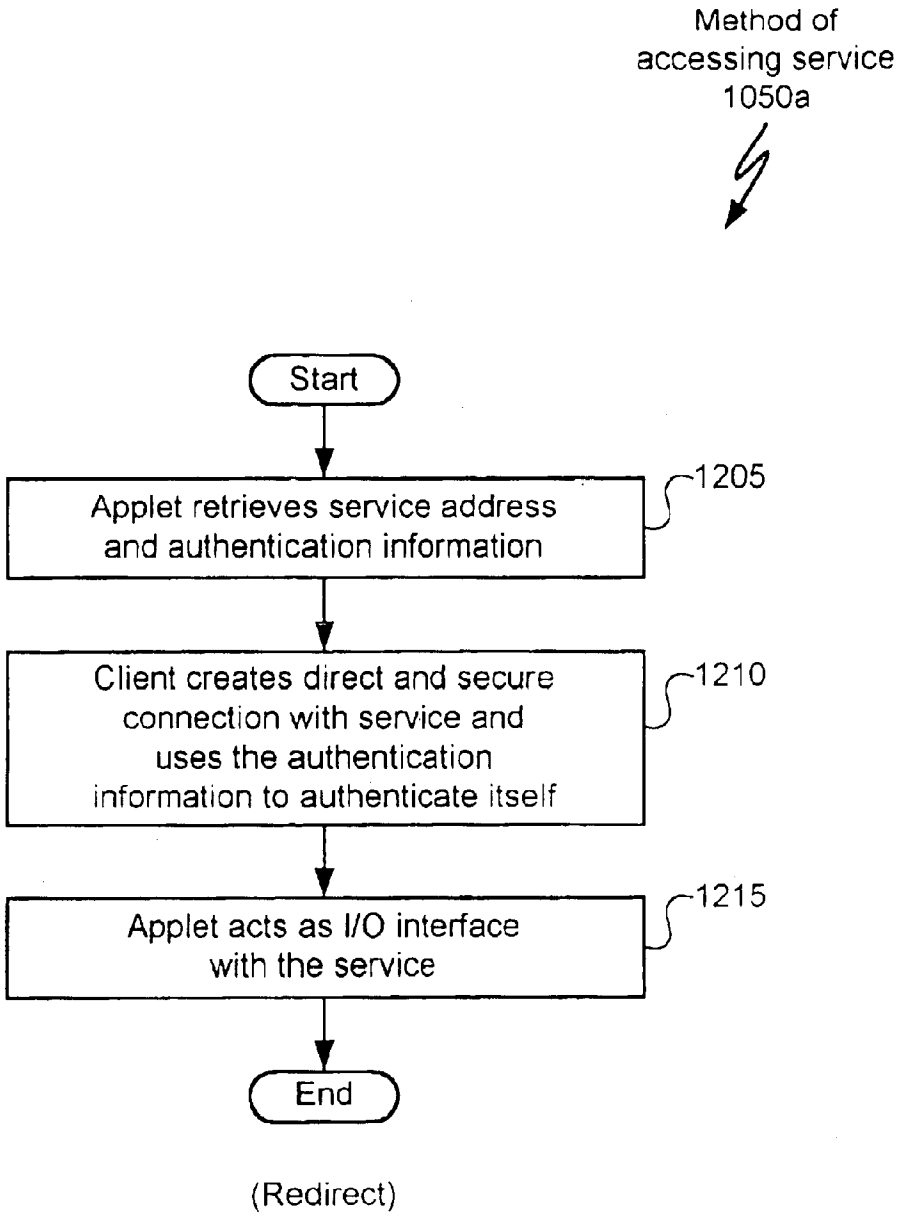

Method of accessing service 1050a
Start
Applet retrieves service address and authentication information
1205
Client creates direct and secure connection with service and uses the authentication information to authenticate itself
1210
Applet acts as I/O interface with the service
1215
End
(Redirect)

FIG. 12

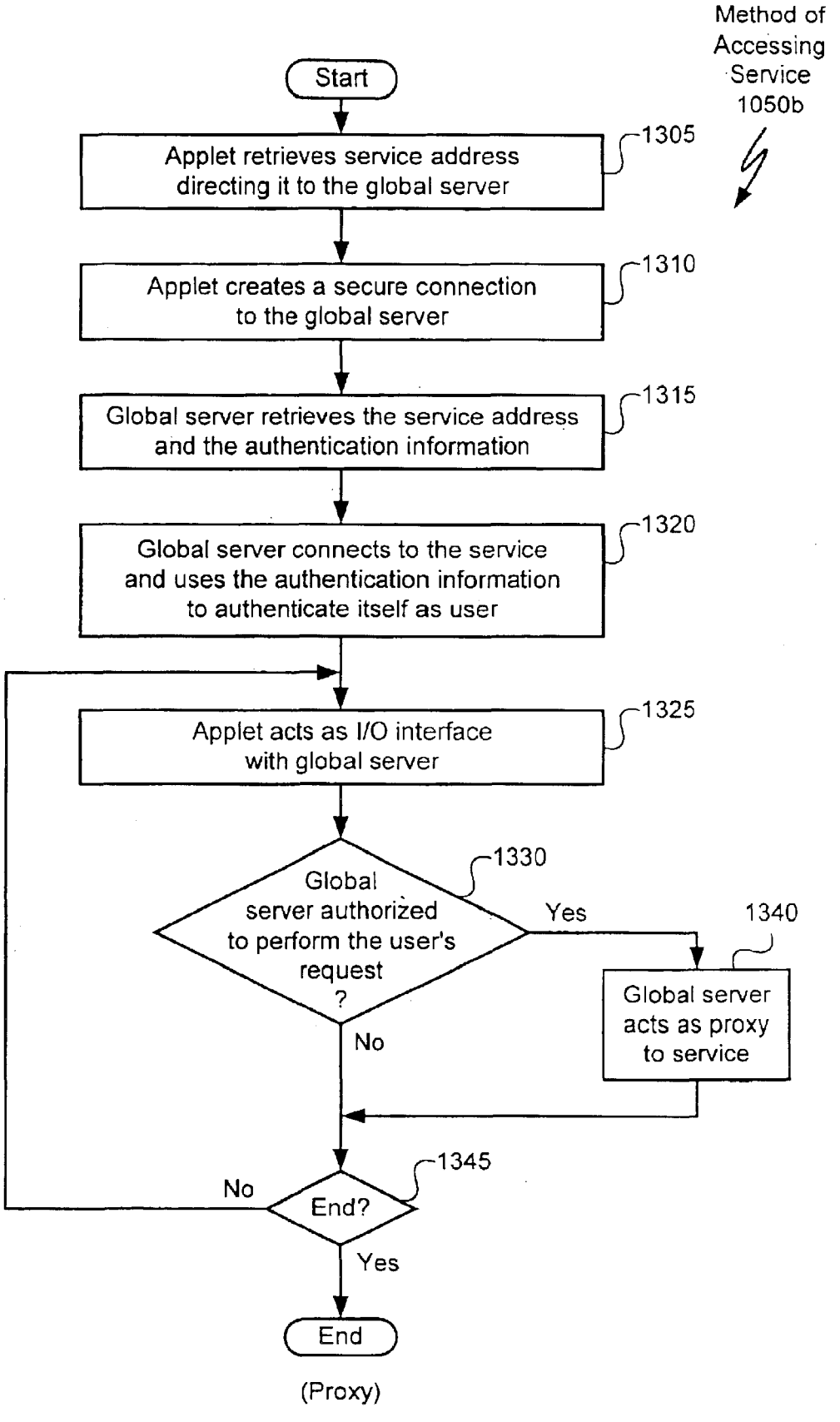

Method of Accessing Service 1050b
Start
Applet retrieves service address directing it to the global server
1305
Applet creates a secure connection to the global server
1310
Global server retrieves the service address and the authentication information
1315
Global server connects to the service and uses the authentication information to authenticate itself as user
1320
Applet acts as I/O interface with global server
1325
Global server authorized to perform the user's request ?
1330
Yes
1340
Global server acts as proxy to service
No
End?
1345
No
Yes
End
(Proxy)

FIG. 13

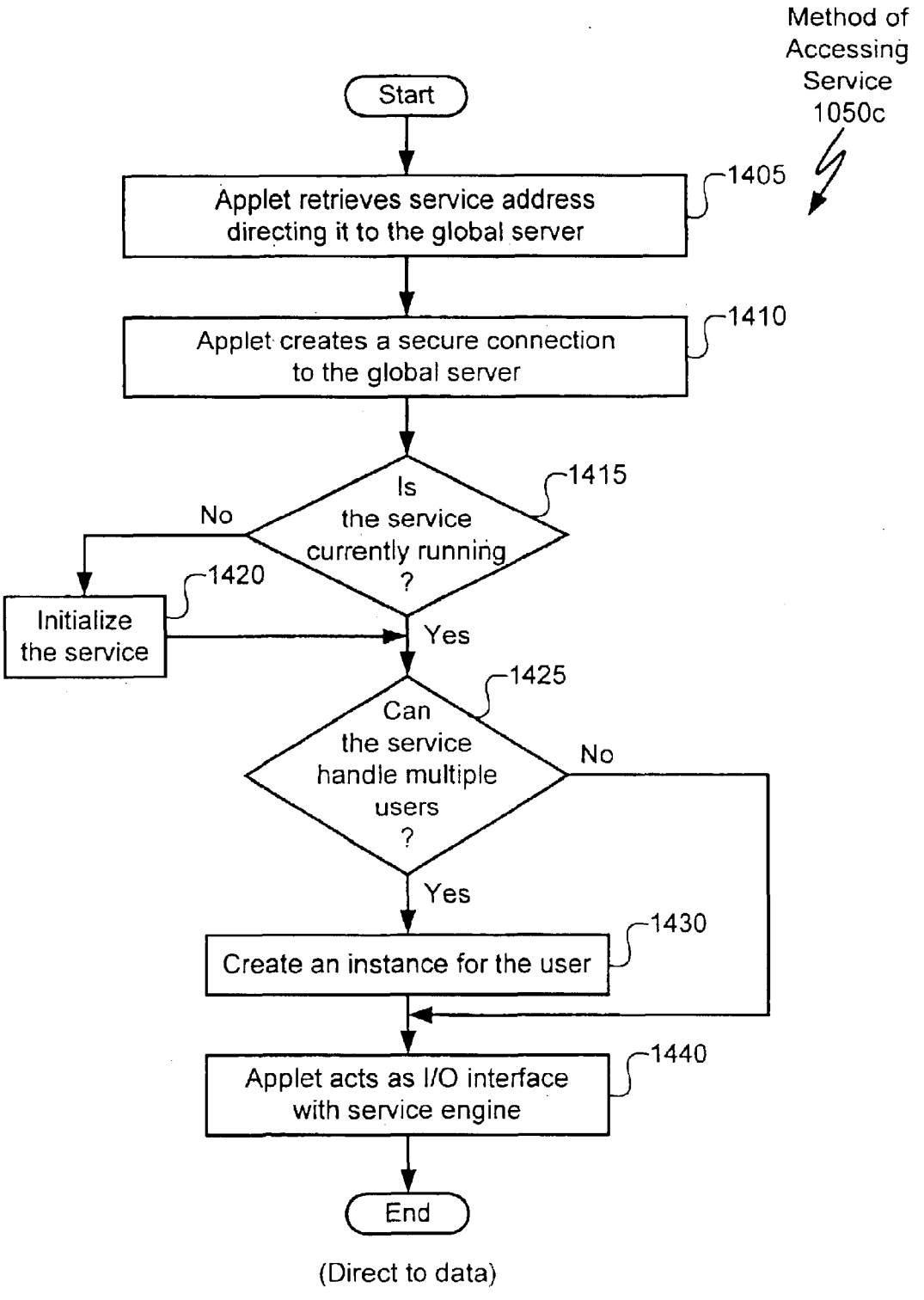

Method of Accessing Service 1050c
Start
Applet retrieves service address directing it to the global server
1405
Applet creates a secure connection to the global server
1410
Is the service currently running ?
1415
No
Initialize the service
1420
Yes
Can the service handle multiple users ?
1425
No
Yes
Create an instance for the user
1430
Applet acts as I/O interface with service engine
1440
End
(Direct to data)

FIG. 14

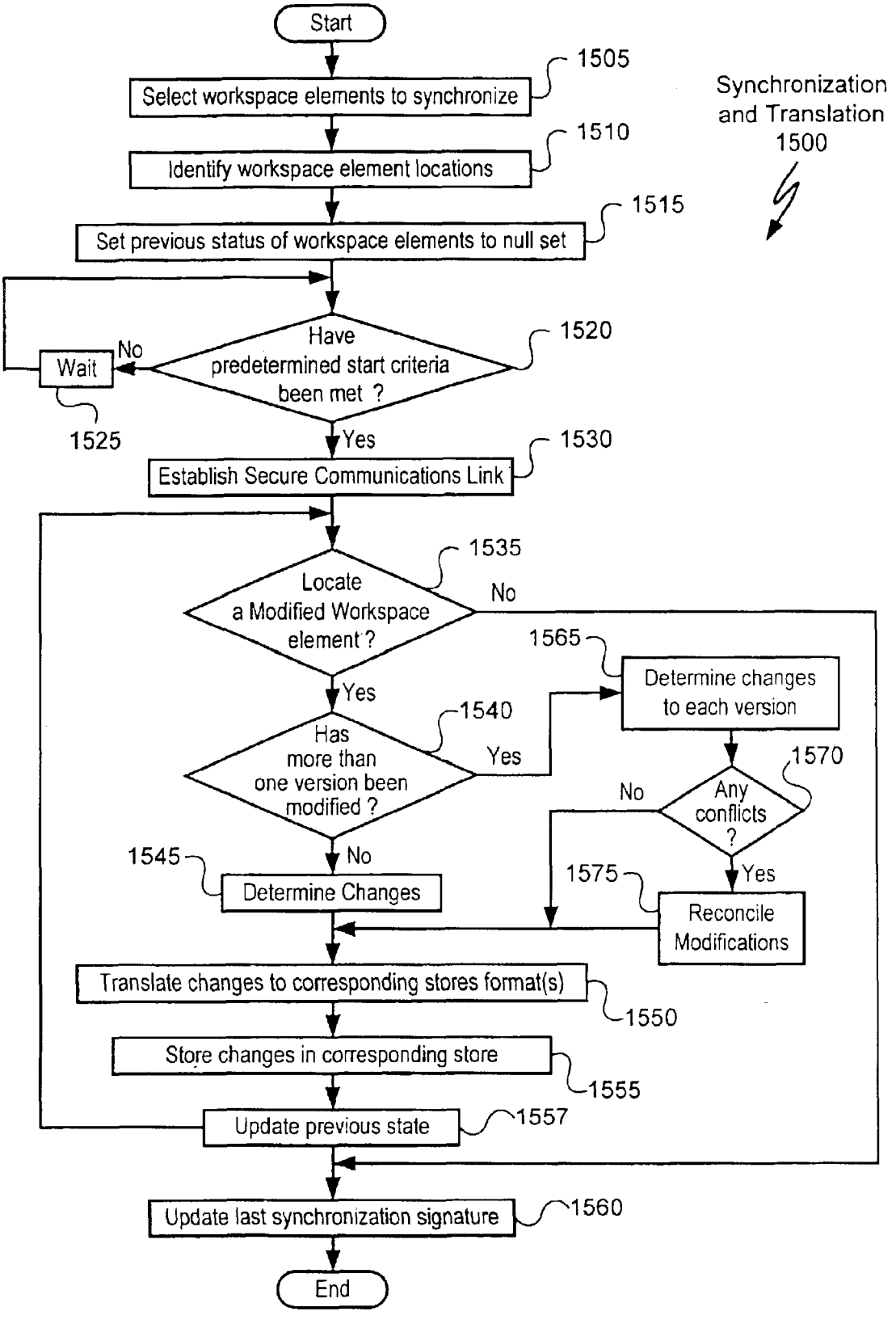

Start
Select workspace elements to synchronize
1505
Identify workspace element locations
1510
Set previous status of workspace elements to null set
1515
Synchronization and Translation 1500
Have predetermined start criteria been met ?
1520
No
Wait
1525
Yes
Establish Secure Communications Link
1530
Locate a Modified Workspace element ?
1535
No
Yes
Has more than one version been modified ?
1540
Yes
No
Determine changes to each version
1565
Any conflicts ?
1570
No
Yes
Determine Changes
1545
Reconcile Modifications
1575
Translate changes to corresponding stores format(s)
1550
Store changes in corresponding store
1555
Update previous state
1557
Update last synchronization signature
1560
End

FIG. 15

# SYSTEM AND METHOD FOR GLOBALLY AND SECURELY ACCESSING UNIFIED INFORMATION IN A COMPUTER NETWORK

## CROSS-REFERNCE TO RELATED APPLICATIONS

This application is a continuation of and incorporates by reference patent application Ser. No. 09/666,877, entitled "System and Method for Globally and Securely Accessing Unified Information in a Computer Network" filed on Sep. 20, 2000, now U.S. Pat. No. 6,708,221, by inventors Daniel J. Mendez, Mark D. Riggins, Prasad Wagle, Hong Q. Bui, Mason Ng. Sean Michael Quinlan, Christine C. Ying, Christopher R. Zuleeg, David J. Cowan, Joanna A. Aptekar-Strober and R. Stanley Bailes, which application is a continuation of and incorporates by reference parent application U.S. patent application Ser. No. 08/903,118 entitled "System and Method for Globally and Securely Accessing Unified Information in a Computer Network" of Daniel J. Mendez, Mark D. Riggins, Prasad Wagle, Hong Q. Bui, Mason Ng, Sean Michael Quinlan, Christine C. Ying, Christopher R. Zuleeg, David J. Cowan, Joanna A. Aptekar-Strober and R. Stanley Bailes, filed Jul. 30, 1997, now abandoned,which is a continuation-in-part of patent application entitled "System and Method for Globally Accessing Computer Services," Ser. No. 08/766,307, now issued as Pat. No. 6,131,116, filed on Dec. 13, 1996, by inventors Mark D. Riggins, R. Stanley Bailes, Hong Q. Bui, David J. Cowan, Daniel J. Mendez, Mason Ng, Sean Michael Quinlan, Prasad Wagle, Christine C. Ying, Christopher R. Zuleeg and Joanna A. Aptekar-Strober; and of co-pending patent application entitled "System and Method for Enabling Secure Access to Services in a Computer Network," Ser. No. 08/841,950, filed on Apr. 8, 1997, by inventor Mark Riggins; and of patent application entitled "System and Method for Securely Synchronizing Multiple Copies of a Workspace Element in a Network," Ser. No. 08/835,997, now issued as Pat. No. 6,085,192, filed on Apr. 11, 1997, by inventors Daniel J. Mendez, Mark D. Riggins, Prasad Wagle and Christine C. Ying; and of patent application entitled "System and Method for Using a Global Translator to Synchronize Workspace Elements Across a Network," Ser. No. 08/865,075, now issued as Pat. No. 6,023,708, filed on May 29, 1997, by inventors Daniel J. Mendez, Mark D. Riggins, Prasad Wagle and Christine C. Ying. These applications have been commonly assigned to Visto Corporation, and are incorporated herein by reference as if copied verbatim hereafter. Benefit of the earlier filing dates is claimed on all common subject matter.

## BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly provides a system and method for globally and securely accessing unified information in a computer network.

2. Description of the Background Art

The internet currently interconnects about 100,000 computer networks and several million computers. Each of these computers stores numerous application programs for providing numerous services, such as generating, sending and receiving e-mail, accessing World Wide Web sites, generating and receiving facsimile documents, storing and retrieving data, etc.

A roaming user, i.e., a user who travels and accesses a workstation remotely, is faced with several problems. Program designers have developed communication techniques for enabling the roaming user to establish a communications link and to download needed information and needed service application programs from the remote workstation to a local computer. Using these techniques, the roaming user can manipulate the data on the remote workstation and, when finished, can upload the manipulated data back from the remote workstation to the local computer. However, slow computers and slow communication channels make downloading large files and programs a time-consuming process. Further, downloading files and programs across insecure channels severely threatens the integrity and confidentiality of the downloaded data.

Data consistency is also a significant concern for the roaming user. For example, when maintaining multiple independently modifiable copies of a document, a user risks using an outdated version. By the time the user notices an inconsistency, interparty miscommunication or data loss may have already resulted. The user must then spend more time attempting to reconcile the inconsistent versions and addressing any miscommunications.

The problem of data inconsistency is exacerbated when multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.

Data consistency problems may also arise when using application programs from different vendors. For example, the Netscape Navigator™ web engine and the Internet Explorer™ web engine each store bookmarks for quick reference to interesting web sites. However, since each web engine uses different formats and stores bookmarks in different files, the bookmarks are not interchangeable. In addition, one web engine may store a needed bookmark, and the other may not. A user who, for example, runs the Internet Explorer™ web engine at home and runs the Netscape Navigator™ web engine at work risks having inconsistent bookmarks at each location.

Therefore, a system and method are needed to enable multiple users to access computer services remotely without consuming excessive user time, without severely threatening the integrity and confidentiality of the data, and without compromising data consistency.

## SUMMARY OF THE INVENTION

The present invention provides a system and methods for providing global and secure access to services and to unified (synchronized) workspace elements in a computer network. A user can gain access to a global server using any terminal, which is connected via a computer network such as the Internet to the global server and which is enabled with a web engine.

A client stores a first set of workspace data, and is coupled via a computer network to a global server. The client is configured to synchronize selected portions of the first set of workspace data (comprising workspace elements) with the global server, which stores independently modifiable copies of the selected portions. The global server may also store workspace data not received from the client, such as e-mail sent directly to the global server. Accordingly, the global server stores a second set of workspace data. The global server is configured to identify and authenticate a user attempting to access it from a remote terminal, and is

configured to provide access based on the client configuration either to the first set of workspace data stored on the client or to the second set of workspace data stored on the global server. It will be appreciated that the global server can manage multiple clients and can synchronize workspace data between clients.

Service engines for managing services such as e-mail management, accessing bookmarks, calendaring, network access, etc. may be stored anywhere in the computer network, including on the client, on the global server or on any other computer. The global server is configured to provide the user with access to services, which based on level of authentication management or user preferences may include only a subset of available services. Upon receiving a service request from the client, the global server sends configuration information to enable access to the service.

Each client includes a base system and the global server includes a synchronization agent. The base system and synchronization agent automatically establish a secure connection therebetween and synchronize the selected portions of the first set of workspace data stored on the client and the second set of workspace data stored on the global server. The base system operates on the client and examines the selected portions to determine whether any workspace elements have been modified since last synchronization. The synchronization agent operates on the global server and informs the base system whether any of the workspace elements in the second set have been modified. Modified version may then be exchanged so that an updated set of workspace elements may be stored at both locations, and so that the remote user can access an updated database. If a conflict exists between two versions, the base system then performs a responsive action such as examining content and generating a preferred version, which may be stored at both locations. The system may further include a synchronization-start module at the client site (which may be protected by a firewall) that initiates interconnection and synchronization when predetermined criteria have been satisfied.

A method of the present invention includes establishing a communications link between the client and the global server. The method includes establishing a communications link between the client and a service based upon user requests. The method receives configuration data and uses the configuration data to configure the client components such as the operating system, the web engine and other components. Configuring client components enables the client to communicate with the service and provides a user-and-service-specific user interface on the client. Establishing a communications link may also include confirming access privileges.

Another method uses a global translator to synchronize workspace elements. The method includes the steps of selecting workspace elements for synchronization, establishing a communications link between a client and a global server, examining version information for each of the workspace elements on the client and on the global server to determine workspace elements which have been modified since last synchronization. The method continues by comparing the corresponding versions and performing a responsive action. Responsive actions may include storing the preferred version at both stores or reconciling the versions using content-based analysis.

The system and methods of the present invention advantageously provide a secure globally accessible third party, i.e. the global server. The system and methods provide a secure technique for enabling a user to access the global server and thus workspace data remotely and securely. Because of the global firewall and the identification and security services performed by the global server, corporations can store relatively secret information on the global server for use by authorized clients. Yet, the present invention also enables corporations to maintain only a portion of their secret information on the global server, so that there would be only limited loss should the global server be compromised. Further, the global server may advantageously act as a client proxy for controlling access to services, logging use of keys and logging access of resources.

A client user who maintains a work site, a home site, an off-site and the global server site can securely synchronize the workspace data or portions thereof among all four sites. Further, the predetermined criteria (which control when the synchronization-start module initiates synchronization) may be set so that the general synchronization module synchronizes the workspace data upon user request, at predetermined times during the day such as while the user is commuting, or after a predetermined user action such as user log-off or user log-on. Because the system and method operate over the Internet, the system is accessible using any connected terminal having a web engine such as an internet-enabled smart phone, television settop (e.g., web TV), etc. and is accessible over any distance. Since the system and method include format translation, merging of workspace elements between different application programs and different platforms is possible. Further, because synchronization is initiated from within the firewall, the typical firewall, which prevents in-bound communications and only some protocols of out-bound communications, does not act as an impediment to workspace element synchronization.

Further, a roaming user may be enabled to access workspace data from the global server or may be enabled to access a service for accessing workspace data from a client. For example, a user may prefer not to store personal information on the global server but may prefer to have remote access to the information. Further, the user may prefer to store highly confidential workspace elements on the client at work as added security should the global server be compromised.

The present invention may further benefit the roaming user who needs emergency access to information. The roaming user may request a Management Information Systems (MIS) director controlling the client to provide the global server with the proper keys to enable access to the information on the client. If only temporary access is desired, the keys can then be later destroyed either automatically or upon request. Alternatively, the MIS director may select the needed information as workspace elements to be synchronized and may request immediate synchronization with the global server. Accordingly, the global server and the client can synchronize the needed information, and the user can access the information from the global server after it has completed synchronization.

The present invention also enables the system and methods to synchronize keys, available services and corresponding service addresses to update accessibility of workspace data and services. For example, if the user of a client accesses a site on the Internet which requires a digital certificate and the user obtains the certificate, the system and methods of the present invention may synchronize this newly obtained certificate with the keys stored on the global server. Thus, the user need not contact the global server to provide it with the information. The synchronization means will synchronize the information automatically.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating a secure data-synchronizing remotely accessible network in accordance with the present invention;

FIG. **2** is a block diagram illustrating details of a FIG. **1** remote terminal;

FIG. **3** is a block diagram illustrating details of a FIG. **1** global server;

FIG. **4** is a block diagram illustrating details of a FIG. **1** synchronization agent;

FIG. **5** is a graphical representation of an example bookmark in global format;

FIG. **6** is a graphical representation of the FIG. **3** configuration data;

FIG. **7** is a block diagram illustrating the details of a FIG. **1** client;

FIG. **8** is a block diagram illustrating the details of a FIG. **1** base system;

FIG.**9** illustrates an example services list;

FIG. **10** is a flowchart illustrating a method for remotely accessing a secure server;

FIG. **11** is a flowchart illustrating details of the FIG. **10** step of creating a link between a client and global server;

FIG. **12** is a flowchart illustrating details of the FIG. **10** step of providing access to a service in a first embodiment;

FIG. **13** is a flowchart illustrating details of the FIG. **10** step of providing access to a service in a second embodiment;

FIG. **14** is a flowchart illustrating details of the FIG. **10** step of providing access to a service in a third embodiment; and

FIG. **15** is a flowchart illustrating a method for synchronizing multiple copies of a workspace element over a secure network.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. **1** is a block diagram illustrating a network **100**, comprising a first site such as a remote computer terminal **105** coupled via a communications channel **110** to a global server **115**. The global server **115** is in turn coupled via a communications channel **120** to a second site such as a Local Area Network (LAN) **125** and via a communications channel **122** to a third site such as client **167**. Communications channel **110**, communications channel **120** and communications channel **122** may be referred to as components of a computer network such as the Internet. The global server **115** is protected by a global firewall. **130**, an d the LAN **125** is protected by a LAN firewall **135**.

The LAN **125** comprises a client **165**, which includes a base system **170** for synchronizing workspace data **180** (e-mail data, file data, calendar data, user data, etc.) with the global server **115** and may include a service engine **175** for providing computer services such as scheduling, e-mail, paging, word-processing or the like. Those skilled in the art will recognize that workspace data **180** may include other types of data such as application programs. It will be further appreciated that workspace data **180** may each be divided into workspace elements, wherein each workspace element may be identified by particular version information **782** (FIG. **7**). For example, each e-mail, file, calendar, etc. may be referred to as "a workspace element in workspace data."



For simplicity, each workspace element on the client **165** is referred to herein as being stored in format A. It will be further appreciated that the workspace data **180** or portions thereof may be stored at different locations such as locally on the client **165**, on other systems in the LAN **125** or on other systems (not shown) connected to the global server **115**.

The client **167** is similar to the client **165**. However, workspace data stored on the client **167** is referred to as being stored in format B, which may be the same as or different than format A. All aspects described above and below with reference to the client **165** are also possible with respect to the client **167**. For example, client **167** may include services (not shown) accessible from remote terminal **105**, may include a base system (not shown) for synchronizing workspace elements with the global server **115**, etc.

The global server **115** includes a security system **160** for providing only an authorized user with secure access through firewalls to services. The security system **160** may perform identification and authentication services and may accordingly enable multiple levels of access based on the level of identification and authentication. The global server **115** further includes a configuration system **155** that downloads configuration data **356** (FIGS. **3** and **6**) to the remote terminal **105** to configure remote terminal **105** components such as the operating system **270** (FIG. **2**), the web engine **283** (FIG. **2**), the applet engine **290** (FIG. **2**), etc. The configuration system **155** uses the configuration data **356** to enable the remote terminal **105** to access the services provided by the service engine **175** and to provide a user-and-service-specific user interface.

The global server **115** stores workspace data **163**., which includes an independently modifiable copy of each selected workspace element in the selected portions of the workspace data **180**. Accordingly, the workspace data **163** includes an independently modifiable copy of each corresponding version information **782** (FIG. **7**). The workspace data **163** may also include workspace elements which originate on the global server **115** such as e-mails sent directly to the global server **115** or workspace elements which are downloaded from another client (not shown). The global server **115** maintains the workspace data **163** in a format, referred to as a "global format," which is selected to be easily translatable by the global translator **150** to and from format A and to and from format B. As with format A and format B, one skilled in the art knows that the global format actually includes a global format for each information type. For example, there may be a global format for bookmarks (FIG. **5**), a global format for files, a global format for calendar data, a global format for e-mails, etc.

The global server **115** also includes a synchronization agent **145** for examining the workspace elements of workspace data **163**. More particularly, the base system **170** and the synchronization agent **145**, collectively referred to herein as "synchronization means," cooperate to synchronize the workspace data **163** with the selected portions of the workspace data **180**. The synchronization means may individually synchronize workspace elements (e.g., specific word processor documents) or may synchronize workspace element folders (e.g., a bookmark folder). Generally, the base system **170** manages the selected portions of the workspace data **180** within the LAN **125** and the synchronization agent **145** manages the selected portions of workspace data **163** within the global server **115**. It will be appreciated that the global translator **150** cooperates with the synchronization means to translate between format A (or

format B) and the global format. It will be further appreciated that the global server **115** may synchronize the workspace data **163** with workspace data **180** and with the workspace data (not shown) on the client **167**. Accordingly, the workspace data **163** can be easily synchronized with the workspace data (not shown) on the client **167**.

The remote terminal **105** includes a web engine **140**, which sends requests to the global server **115** and receives information to display from the global server **115**. The web engine **140** may use HyperText Transfer Protocol (HTTP) and HyperText Markup Language (HTML) to interface with the global server **115**. The web engine **140** may be enabled to run applets, which when executed operate as the security interface for providing access to the global server **115** and which operate as the application interface with the requested service. Using the present invention, a user can operate any remote client **105** connected to the Internet to access the global server **115**, and thus to access the services and the workspace data on or accessible by the global server **115**.

FIG. **2** is a block diagram illustrating details of the remote terminal. **105**, which includes a Central Processing Unit (CPU) **210** such as a Motorola Power PC™ microprocessor or an Intel Pentium™ microprocessor. An input device **220** such as a keyboard and mouse, and an output device **230** such as a Cathode Ray Tube (CRT) display are coupled via a signal bus **235** to CPU **210**. A communications interface **240**, a data storage device **250** such as Read Only Memory (ROM) and a magnetic disk, and a Random-Access Memory (RAM) **260** are further coupled via signal bus **235** to CPU **210**. The communications interface **240** is coupled to a communications channel **110** as shown in FIG. **1**.

An operating system **270** includes a program for controlling processing by CPU **210**, and is typically stored in data storage device **250** and loaded into RAM **260** (as shown) for execution. Operating system **270** further includes a communications engine **275** for generating and transferring message packets via the communications interface **240** to and from the communications channel **110**. Operating system **270** further includes an Operating System (OS) configuration module **278**, which configures the operating system **270** based on OS configuration data **356** (FIG. **3**) such as Transmission Control Protocol (TCP) data, Domain Name Server (DNS) addresses, etc. received from the global server **115**.

Operating system **270** further includes the web engine **140** for communicating with the global server **115**. The web engine **140** may include a web engine (WE) configuration module **286** for configuring elements of the web engine **140** such as home page addresses, bookmarks, caching data, user preferences, etc. based on the configuration data **356** received from the global server **115**. The web engine **140** may also include an encryption engine **283** for using encryption techniques to communicate with the global server **115**. The web engine **140** further may include an applet engine **290** for handling the execution of downloaded applets including applets for providing security. The applet engine **290** may include an Applet Engine (AE) configuration module **295** for configuring the elements of the applet engine **290** based on configuration data **356** received from the global server **115**.

FIG. **3** is a block diagram illustrating details of the global server. **115**, which includes a Central Processing Unit (CPU) **310** such as a Motorola Power PC™ microprocessor or an Intel Pentium™ microprocessor. An input device **320** such as a keyboard and mouse, and an output device **330** such as a Cathode Ray Tube (CRT) display are coupled via a signal bus **335** to CPU **310**. A communications interface **340**, a data storage device **350** such as Read Only Memory (ROM) and a magnetic disk, and a Random-Access Memory (RAM) **370** are further coupled via signal bus **335** to CPU **310**. As shown in FIG. **1**, the communications interface **340** is coupled to the communications channel **110** and to the communications channel **120**.

An operating system **380** includes a program for controlling processing by CPU **310**, and is typically stored in data storage device **359** and loaded into RAM **370** (as illustrated) for execution. The operating system **380** further includes a communications engine **382** for generating and transferring message packets via the communications interface **340** to and from the communications channel **345**. The operating system **380** also includes a web page engine **398** for transmitting web page data **368** to the remote terminal **105**, so that the remote terminal **105** can display a web page **900** (FIG. **9**) listing functionality offered by the global server **115**. Other web page data **368** may include information for displaying security method selections.

The operating system **380** may include an applet host engine **395** for transmitting applets to the remote terminal **105**. A configuration engine **389** operates in conjunction with the applet host engine **395** for transmitting configuration applets **359** and configuration and user data **356** to the remote terminal **105**. The remote terminal **105**. executes the configuration applets **359** and uses the configuration and user data **356** to configure the elements (e.g., the operating system **270**, the web engine **140** and the applet engine **290**) of the remote terminal **105**. Configuration and user data **356** is described in greater detail with reference to FIG. **6**.

The operating system **380** also includes the synchronization agent **145** described with reference to FIG. **1**. The synchronization agent **145** synchronizes the workspace data **163** on the global server **115** with the workspace data **180** on the client **165**. As stated above with reference to FIG. **1**, the global translator **150** translates between format A used by the client **165** and the global format used by the global server **115**.

The operating system **380** may also includes a security engine **392** for determining whether to instruct a communications engine **382** to create a secure communications link with a client **165** or terminal **105**, and for determining the access rights of the user. For example, the security engine **392** forwards to the client **165** or remote terminal **105** security applets **362**, which when executed by the receiver poll the user and respond back to the global server **115**. The global server **115** can examine the response to identify and authenticate the user.

For example, when a client **165** attempts to access the global server **115**, the security engine **384** determines whether the global server **115** accepts in-bound communications from a particular port. If so, the security engine **392** allows the communications engine **382** to open a communications channel **345** to the client **165**. Otherwise, no channel will be opened. After a channel is opened, the security engine **392** forwards an authentication security applet **362** to the remote terminal **105** to poll the user for identification and authentication information such as for a user ID and a password. The authentication security applet **362** will generate and forward a response back to the global server **115**, which will use the information to verify the identity of the user and provide access accordingly.

It will be appreciated that a "request-servicing engine" may be the configuration engine **389** and the applet host engine **395** when providing services to a remote terminal

105 or client 165. The request-servicing engine may be the web page engine 398 when performing workspace data 163 retrieval operations directly from the global server 115. The request-servicing engine may be the configuration engine 389 and the applet host engine 395 when performing workspace data 180 retrieval operations from the client 165 or from any other site connected to the global server 115. The request-servicing engine may be security engine 392 when performing security services such as user identification and authentication. The request-servicing engine May be the synchronization agent when the performing synchronization with the client 165. Further, the request-servicing engine may be any combination of these components.

FIG. 4 is a block diagram illustrating details of the synchronization agent 145, which includes a communications module 405 and a general synchronization module 410. The communications module 405 includes routines for compressing data and routines for communicating via the communications channel 120 with the base system 170. The communications module 405 may further include routines for communicating securely channel through the global firewall 130 and through the LAN firewall 125.

The general synchronization module 410 includes routines for determining whether workspace elements have been synchronized and routines for forwarding to the base system 170 version information (not shown) of elements determined to be modified after last synchronization. The general synchronization module 410 may either maintain its own last synchronization signature (not shown), receive a copy of the last synchronization signature with the request to synchronize from the base system 170, or any other means for insuring that the workspace data has been synchronized. The general synchronization module 410 further includes routines for receiving preferred versions of workspace data 180 workspace elements from the base system 170 and routines for forwarding preferred versions of workspace data 180 workspace elements to the base system 170.

FIG. 5 illustrates an example bookmark workspace element in the global format. The translator 150 incorporates all the information needed to translate between all incorporated formats. For example, if for a first client a bookmark in format A needs elements X, Y and Z and for a second client a bookmark in format B needs elements W, X and Y, the global translator 150 incorporates elements W, X, Y and Z to generate a bookmark in the global format. Further, the translator 150 incorporates the information which is needed by the synchronization means (as described below in FIG. 4) such as the last modified date. Accordingly, a bookmark in the Global Format may include a user identification (ID) 505, an entry ID 510, a parent ID 515, a folder ID flag 520, a name 525, a description 530, the Uniform Resource Locator (URL) 535, the position 540, a deleted ID flag 545, a last modified date 550, a created date 555 and a separation ID flag 560.

FIG. 6 is a block diagram illustrating details of the configuration and user data 356. Configuration data 356 includes settings 605 such as TCP data and the DNS address, web browser settings such as home page address, bookmarks and caching data, applet engine settings, and applet configuration data such as the user's e-mail address, name and signature block. It will be appreciated that applet-specific configuration and user data 356 is needed, since the service may not be located on the user's own local client 165. Configuration and user data 356 further includes predetermined user preferences 610 such as font, window size, text size, etc.

Configuration data 356 further includes the set of services 615, which will be provided to the user. Services 615 include a list of registered users and each user's list of user-preferred available services 615. Services may also include a list of authentication levels needed to access the services 615. Configuration and user data 137 further includes service addresses 620 specifying the location of each of the services 615 accessible via the global server 115.

FIG. 7 is a block diagram illustrating details of the client 165, which includes a CPU 705, an input device 710, an output device 725, a communications interface 710, a data storage device 720 and RAM 730, each coupled to a signal bus 740.

An operating system 735 includes a program for controlling processing by the CPU 705, and is typically stored in the data storage device 720 and loaded into the RAM 730 (as illustrated) for execution. A service engine 175 includes a service program for managing workspace data 180 that includes version information (not shown). The service engine 175 may be also stored in the data storage device 720 and loaded into the RAM 730 (as illustrated) for execution. The workspace data 180 may be stored in the data storage device 330. As stated above with reference to FIG. 1, the base system 170 operates to synchronize the workspace data 180 on the client 165 with the workspace data 163 on the global server 115. The base system 170 may be also stored in the data storage device 720 and loaded into the RAM 730 (as shown) for execution. The base system 170 is described in greater detail with reference to FIG. 8.

FIG. 8 is a block diagram illustrating details of the base system 170, which includes a communications module 805, a user interface module 810, locator modules 815, a synchronization-start ("synch-start") module 820, a general synchronization module 825 and a content-based synchronization module 830. For simplicity, each module is illustrated as communicating with one another via a signal bus 840. It will be appreciated that the base system 170 includes the same components as included in the synchronization agent 145.

The communications module 805 includes routines for compressing data, and routines for communicating via the communications interface 710 (FIG. 7) with the synchronization agent 145 (FIG. 1). The communications module 805 may include routines for applying Secure Socket Layer (SSL) technology and user identification and authentication techniques (i.e., digital certificates) to establish a secure communication channel through the LAN firewall 135 and through the global firewall 130. Because synchronization is initiated from within the LAN firewall 135 and uses commonly enabled protocols such as HyperText Transfer Protocol (HTTP), the typical firewall 135 which prevents in-bound communications in general and some outbound protocols does not act as an impediment to e-mail synchronization. Examples of communications modules 805 may include TCP/IP stacks or the AppleTalk™ protocol.

The user interface 810 includes routines for communicating with a user, and may include a conventional Graphical User Interface (GUI). The user interface 810 operates in coordination with the client 165 components as described herein.

The locator modules 815 include routines for identifying the memory locations of the workspace elements in the workspace data 180 and the memory locations of the workspace elements in the workspace data 163. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue boxes to query a user. It will be appreciated that the locator

modules **815** may perform workspace element memory location identification upon system boot-up or after each communication with the global server **115** to maintain updated memory locations of workspace elements.

The synchronization-start module **820** includes routines for determining when to initiate synchronization of workspace data **163** and workspace data **180**. For example, the synchronization-start module **820** may initiate data synchronization upon user request, at a particular time of day, after a predetermined time period passes, after a predetermined number of changes, after a user action such as user log-off or upon like criteria. The synchronization-start module **820** initiates data synchronization by instructing the general synchronization module **825** to begin execution of its routines. It will be appreciated that communications with synchronization agent **145** preferably initiate from within the LAN **125**, because the typical LAN firewall **125** prevents in-bound communications and allows out-bound communications.

The general synchronization module **825** includes routines for requesting version information from the synchronization agent **145** (FIG. **1**) and routines for comparing the version information against a last synchronization signature **835** such as a last synchronization date and time to determine which versions have been modified. The general synchronization module **825** further includes routines for comparing the local and remote versions to determine if only one or both versions of a particular workspace element have been modified and routines for performing an appropriate synchronizing responsive action. Appropriate synchronizing responsive actions may include forwarding the modified version (as the preferred version) of a workspace element in workspace data **180** or forwarding just a compilation of the changes to the other store(s). Other appropriate synchronizing responsive actions may include, if reconciliation between two modified versions is needed, then instructing the content-based synchronization module **830** to execute its routines (described below).

It will be appreciated that the synchronization agent **145** preferably examines the local version information **124** and forwards only the elements that have been modified since the last synchronization signature **835**. This technique makes efficient use of processor power and avoids transferring unnecessary data across the communications channel **712**. The general synchronization module **825** in the LAN **135** accordingly compares the data elements to determine if reconciliation is needed. Upon completion of the data synchronization, the general synchronization module **825** updates the last synchronization signature **835**.

The content-based synchronization module **830** includes routines for reconciling two or more modified versions of workspace data **163**, **180** in the same workspace element. For example, if the original and the copy of a user workspace element have both been modified independently since the last synchronization, the content-based synchronization module **830** determines the appropriate responsive action. The content-based synchronization module **830** may request a user to select the preferred one of the modified versions or may respond based on preset preferences, i.e., by storing both versions in both stores or by integrating the changes into a single preferred version which replaces each modified version at both stores. When both versions are stored at both stores, each version may include a link to the other version so that the user may be advised to select the preferred version.

It will be appreciated that any client **165** that wants synchronization may have a base system **170**. Alternatively, one base system **170** can manage multiple clients **165**. It will be further appreciated that for a thin client **165** of limited computing power such as a smart telephone, all synchronization may be performed by the global server **115**. Accordingly, components of the base system **170** such as the user interface module **810**, the locator modules **815**, the general synchronization module **825** and the content-based synchronization module **830** may be located on the global server **115**. To initiate synchronization from the client **165**, the client **165** includes the communications module **805** and the synch-start module **820**.

FIG. **9** illustrates an example list **900** of accessible services provided by a URL-addressable HyperText Markup Language (HTML)-based web page, as maintained by the web page engine **398** of the global server **115**. The list **900** includes a title **910** "Remote User's Home Page," a listing of the provided services **615** and a pointer **970** for selecting one of the provided services **615**. As illustrated, the provided services may include an e-mail service **920**, a calendaring service **930**, an internet access service **940**, a paging service **950**, a fax sending service **960**, a user authentication service **963** and a workspace data retrieval service **967**. Although not shown, other services **615** such as bookmarking, QuickCard™, etc. may be included in the list **900**. Although the web page provides the services **615** in a list **900**, other data structures such as a pie chart or table may alternatively be used.

FIG. **10** is a flowchart illustrating a method **1000** for enabling a user to access the services **615** in the computer network system **100**. Method **1000** begins by the remote terminal **105** in step **1005** creating a communications link with the global server **115**. The global server **115** in step **1010** confirms that the user has privileges to access the functionality of the global server **115**. Confirming user access privileges may include examining a user certificate, obtaining a secret password, using digital signature technology, performing a challenge/response technique, etc. It will be appreciated that the security engine **392** may cause the applet host engine **395** to forward via the communications channel **345** to the remote terminal **105** an authentication security applet **362** which when executed communicates with the global server **115** to authenticate the user.

After user access privileges are confirmed, the web page engine **398** of the global server **115** in step **1015** transmits web page data **368** and configuration and user data **356** to the remote terminal **105**. The web engine **140** of the remote terminal **105** in step **1020** uses the web page data **368** and the configuration and user data **356** to display a web page service list **900** (FIG. **9**) on the output device **230**, and to enable access to the services **615** which the global server **115** offers. An example service list **900** is shown and described with reference to FIG. **9**. Configuration of the remote terminal **105** and of the web page **700** is described in detail in the cross-referenced patent applications.

From the options listed on the web page **900**, the user in step **1025** selects a service **615** via input device **220**. In response, the request-servicing engine (described with reference to FIG. **3**) provides the selected service **615**. For example, the applet host engine **395** of the global server **115** in step **1030** may download to the remote terminal **105** a corresponding applet **359** and configuration and user data **356** for executing the requested service **615**. Alternatively, the web page engine **398** may use, for example, HTTP and HTML to provide the selected service **615**. As described above with reference to FIG. **6**, the configuration and user data **356** may include user-specific preferences such as user-preferred fonts for configuring the selected service **615**.

Configuration and user data **356** may also include user-specific and service-specific-information such as stored bookmarks, calendar data, pager numbers, etc. Alternatively, the corresponding applet **359** and the configuration and user data **356** could have been downloaded in step **1015**. Providing access to the service by an applet **359** is described in greater detail below with reference to FIGS. **12**–**14**.

The applet engine **290** of the remote terminal **105** in step **1035** initiates execution of the corresponding downloaded applet. The global server **115** in step **1040** initiates the selected service **615** and in step **1045** selects one of three modes described with reference to FIGS. **12**–**14** for accessing the service **615**. For example, if the user selects a service **615** on a service server (e.g., the client **165**) that is not protected by a separate firewall, then the global server **115** may provide the user with direct access. If the user selects a service **615** provided by a service server within the LAN **125**, then the global server **115** may access the service **615** as a proxy for the user. It will be appreciated that each firewall **130** and **135** may store policies establishing the proper mode of access the global server **115** should select. Other factors for selecting mode of access may include user preference, availability and feasibility. The global server **115** in step **1050** uses the selected mode to provide the remote terminal **105** user with access to the selected service **615**.

FIG. **11** is a flowchart illustrating details of step **1005**, which begins by the remote terminal **105** in step **1105** using a known Uniform Resource Locator (URL) to call the global server **115**. The global server **115** and the remote terminal **105** in step **1107** create a secure communications channel therebetween, possibly by applying Secure Sockets Layer (SSL) technology. That is, the security engine **392** of the global server **115** in step **1110** determines if in-bound secure communications are permitted and, if so, creates a communications channel with the remote terminal **105**. The web engine **140** of the remote terminal **105** and the security engine **392** of the global server **115** in step **1115** negotiate secure communications channel parameters, possibly using public key certificates. An example secure communications channel is RSA with RC**4** encryption. Step **1115** thus may include selecting an encryption protocol which is known by both the global server **115** and the remote terminal **105**. The encryption engine **283** of the remote terminal **105** and secure communications engine **392** of the global server **115** in step **1120** use the secure channel parameters to create the secure communications channel. Method **505** then ends.

FIG. **12** is a flowchart illustrating details of step **1050** in a first embodiment, referred to as step **1050***a*, wherein the global server **115** provides the remote terminal **105** with a direct connection to a service **615**. Step **1050***a* begins by the applet engine **290** in step **1205** running a configuration applet **359** for the selected service **615** that retrieves the service address **620** from data storage device **380** and the authentication information from the keysafe **365**. The communications interface **340** in step **1210** creates a direct and secure connection with the communications interface **340** of the global server **115** at the retrieved service address **620**, and uses the authentication information to authenticate itself. The applet in step **1215** acts as the I/O interface with the service **615**. Step **1050***a* then ends.

FIG. **13** is a flowchart illustrating details of step **1050** in a second embodiment, referred to as step **1050***b*, wherein the global server **115** acts for the remote terminal **105** as a proxy to the service **615**. Step **1050***b* begins with a configuration applet **359** in step **1305** requesting the service address **620** for the selected service **615**, which results in retrieving the service address **620** directing the applet **359** to the global server **115**. The applet **359** in step **1310** creates a connection with-communications interface **340** of the global server **115**. The global server **115** in step **1315** retrieves the service address **620** of the selected service **615** and the authentication information for the selected service **615** from the keysafe **365**. The communications interface **340** of the global server **115** in step **1320** negotiates secure channel parameters for creating a secure channel with the service server **1014**. The communications interface **340** in step **1320** also authenticates itself as the user.

Thereafter, the applet **359** in step **1325** acts as the I/O interface with the communications interface **340** of the global server **115**. If the global server **115** in step **1330** determines that it is unauthorized to perform a remote terminal **105** user's request, then the global server **115** in step **1345** determines whether the method **1050***b* ends, e.g., whether the user has quit. If so, then method **1050***b* ends. Otherwise, method **1050***b* returns to step **1325** to obtain another request. If the global server **115** in step **1330** determines that it is authorized to perform the remote terminal **105** user's request, then the global server **115** in step. **1340** acts as the proxy for the remote terminal **105** to the service **615**. As proxy, the global server **115** forwards the service request to the selected service **615** and forwards responses to the requesting applet **359** currently executing on the remote terminal **105**. Method **1050***b* then jumps to step **1345**.

FIG. **14** is a flowchart illustrating details of step **1050** in a third embodiment, referred to as step **1050***c*, wherein the service **615** being requested is located on the global server **115**. Step **1050** begins with an applet in step **1405** retrieving the service address **620** for the selected service **615**, which results in providing the configuration applet **359** with the service address **620** of the service **615** on the global server **115**. Thus, the applet in step **1410** creates a secure connection with the global server **115**. No additional step of identification and authentication is needed since the remote terminal **105** has already identified and authenticated itself to the global server **115** as described with reference to step **1010** of FIG. **10**.

In step **1415**, a determination is made whether the service **615** is currently running. If so, then in step **1425** a determination is made whether the service **615** can handle multiple users. If so, then the global server **115** in step **1430** creates an instance for the user, and the applet in step **1440** acts as the I/O interface with the service **615** on the global server **115**. Method **1050***c* then ends. Otherwise, if the service **615** in step **1425** determines that it cannot handle multiple users, then method **1050***c* proceeds to step **1440**. Further, if in step **1415** the global server **115** determines that the service **615** is not currently running, then the global server **115** in step **1420** initializes the service **615** and proceeds to step **1425**.

FIG. **15** is a flowchart illustrating a method **1500** for using a global translator **150** to synchronize workspace data **163** and workspace data **180** in a secure network **100**. Method **1500** begins with the user interface **900** in step **1505** enabling a user to select workspace elements of workspace data **163** and workspace data **180** for the synchronization means to synchronize. The locator modules **815** in step **1510** identify the memory locations of the workspace elements in workspace data **163** and workspace data **180**. If a selected workspace element does not have a corresponding memory location, such as in the case of adding new workspace elements to the global server **115**, then one is selected. The selected memory location may be a preexisting workspace element or a new workspace element. As stated above,

workspace element memory location identification may be implemented using intelligent software or dialogue boxes. The general synchronization module **825** in step **1515** sets the previous status of the workspace elements equal to the null set, which indicates that all information of the workspace element has been added.

The synchronization-start module **820** in step **1520** determines whether predetermined criteria have been met which indicate that synchronization of the workspace elements selected in step **1505** should start. If not, then the synchronization-start module **820** in step **1525** waits and loops back to step **1520**. Otherwise, the communications module **805** and the communications module **405** in step **1530** establish a secure communications channel therebetween.

The general synchronization module **825** in step **1535** determines whether any workspace elements have been modified. That is, the general synchronization module **825** in step **1535** examines the version information of each selected workspace element in the workspace data **180** against the last synchronization signature **435** to locate modified workspace elements. This comparison may include comparing the date of last modification with the date of last synchronization, or may include a comparison between the current status and the previous status as of the last interaction. Similarly, the general synchronization module **815** examines the version information of each corresponding workspace element in workspace data **163** and the last synchronization signature **435** to locate modified workspace elements.

If in step **1535** no modified workspace elements or folders are located, then the general synchronization module **825** in step **1560** updates the last synchronization signature **435** and method **1500** ends. Otherwise, the general synchronization module **825** in step **1540** determines whether more than one version of a workspace element has been modified since the last synchronization.

If only one version has been modified, then the corresponding general synchronization module **825** in step **1545** determines the changes made. As stated above, determining the changes made may be implemented by comparing the current status of the workspace element against the previous status of the workspace element as of the last interaction therebetween. If the changes were made only to the version in the workspace data **163**, then the global translator **150** in step **1550** translates the changes to the format used by the other store, and the general synchronization module **410** in step **1555** forwards the translated changes to the general synchronization module **825** for updating the outdated workspace element in the workspace data **180**. If the updated version is a workspace element in the workspace data **180**, then the general synchronization module **825** sends the changes to the updated version to the global translator **150** for translation and then to the general synchronization module **410** for updating the outdated workspace element in the workspace data **163**. The general synchronization module **825** and the general synchronization module **410** in step **1557** update the previous state of the workspace element to reflect the current state as of this interaction. Method **1500** then returns to step **1535**.

If the general synchronization module **825** in step **1540** determines that multiple versions have been modified, then the general synchronization module **825** in step **1565** computes the changes to each version and in step **1570** instructs the content-based synchronization module **830** to examine content to determine if any conflicts exist. For example, the content-based synchronization module **830** may determine that a conflict exists if a user deletes a paragraph in one version and modified the same paragraph in another version. The content-based synchronization module **830** may determine that a conflict does not exist if a user deletes different paragraphs in each version. If no conflict is found, then method **1500** jumps to step **1550** for translating and forwarding the changes in each version to the other store. However, if a conflict is found, then the content-based synchronization module **830** in step **1575** reconciles the modified versions. As stated above, reconciliation may include requesting instructions from the user or based on previously selected preferences performing responsive actions such as storing both versions at both stores. It will be appreciated that a link between two versions may be placed in each of the two versions, so that the user will recognize to examine both versions to select the preferred version. Method **1500** then proceeds to step **1550**.

It will be further appreciated that in step **1510** new workspace elements and preexisting workspace elements to which new workspace elements will be merged are set to "modified" and the previous status is set to the null set. Thus, the general synchronization module **825** in step **1540** will determine that more that one version has been modified and the content-based synchronization module **830** in step **1570** will determine that no conflict exists. The changes in each will be translated and forwarded to the other store. Accordingly, the two versions will be effectively merged and stored at each store.

For example, if a first bookmark folder was created by the web engine **140** on the client **165**, a second folder was created by a web engine **140** on the remote terminal **105**, no preexisting folder existed on the global server **115** and the user selected each of these folders for synchronization, then the synchronization means will effectively merge the first and second folders. That is, the general synchronization module **825** on the client **165**, will determine that the first folder has been modified and the previous status is equal to the null set. The general synchronization module **825** will determine and send the changes, i.e., all the workspace elements in the first folder, to a new global folder on the global server **115**. Similarly the general synchronization module (not shown) on the remote terminal **105** will determine that, as of its last interaction, the previous status of each of the second and the global folders is the null set. The general synchronization module **825** will instruct the content-based synchronization module **830** to examine the changes made to each folder to determine whether a conflict exists. Since no conflicts will exist, the general synchronization module **825** will forward the changes to the global folder and the general synchronization module **410** will forward its changes to the second store, thereby merging the workspace elements of the first and second folders in the global and second folders. The general synchronization module **410** will inform the general synchronization module **825** that the global folder has been modified relative to the last interaction, and will forward the new changes to the first folder. Thus, the first and second folders will be merged and stored at each store.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the present invention. For example, a server can be any computer which is polled by a client. Thus, the remote terminal **105** may be referred to as a type of client. Although the system and method have been described with reference to applets, other downloadable executables such as

Java™ applets, Java™ applications or ActiveX™ control developed by the Microsoft Corporation can alternatively be used. Components of this invention may be implemented using a programmed general-purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The invention is limited only by the following claims.

What is claimed is:

**1**. An e-mail system for providing synchronized communication of independently modifiable e-mails over an Internet between a local area network (LAN) server secured by a LAN firewall with at least one normally open LAN firewall port, and each of a plurality of smart-phone devices, said system comprising:

a global server secured by a global server firewall having a global server firewall port therein;

a first Internet communication channel coupling said LAN server to said global server through said open LAN firewall port and said global server firewall port;

a plurality of second Internet communication channels, each coupling said global server to a respective one of said smart-phone devices;

at least one translator for translating e-mail data of different formats such that e- mails transmitted to said global server and said smart-phone devices are of a format or formats which are acceptable thereto;

at least one storage device for storing version information indicating differences between independently modifiable e-mails;

a general synchronization module responsive to a synchronization start command to synchronize different independently modifiable e-mails; and

a synchronization-start module coupled to said general synchronization module, said synchronization-start module being responsive to an existence of predetermined criteria to produce and send a synchronization start command to said general synchronization module.

**2**. A system according to claim **1** wherein the normally open port is an HTTP port.

**3**. A system, according to claim **1**, wherein the normally open port is an HTTPS (SSL).

**4**. A system, according to claim **1**, wherein said storage device is located at the LAN server.

**5**. A system, according to claim **1**, wherein said LAN includes a client device and wherein said storage device is located at said client device.

**6**. A system, according to claim **1**, wherein said storage device is located at said global server.

**7**. A system, according to claim **1**, wherein said storage device is located at one or more of said plurality of said smart-phone devices.

**8**. A system, according to claim **1**, wherein said translator is located at said LAN server.

**9**. A system, according to claim **1**, wherein said LAN includes a client device and wherein said translator is located at said client device.

**10**. A system, according to claim **1**, wherein said translator is located at said global server.

**11**. A system, according to claim **1**, wherein said translator is located at one or more of said plurality of said smart-phone devices.

**12**. A system, according to claim **1**, wherein said general synchronization module is located at said LAN server.

**13**. A system, according to claim **1**, wherein said LAN includes a client device and wherein said general synchronization module is located at said client device.

**14**. A system, according to claim **1**, wherein said general synchronization module is located at said global server.

**15**. A system, according to claim **1**, wherein said general synchronization module is located at one or more of said plurality of said smart-phone devices.

**16**. A system, according to claim **1**, wherein said synchronization-start module is located at said LAN server.

**17**. A system, according to claim **1**, wherein said LAN includes a client device and wherein said synchronization-start module is located at said client device.

**18**. A system, according to claim **1**, wherein said synchronization-start module is located at one or more of said plurality of said smart-phone devices.

* * * * *

# EXHIBIT C

IN THE UNITED STATES DISTRICT COURT

FOR THE EASTERN DISTRICT OF TEXAS

MARSHALL DIVISION

| | | |
|---|---|---|
| SEVEN NETWORKS, INC. | § | |
| Vs. | § | CIVIL ACTION NO. 2:05-CV-365 |
| VISTO CORPORATION | § | |

**ORDER**

Visto's motion for leave to file an amended answer and counterclaims (#43) is granted. Despite Seven's arguments to the contrary, this court concludes that it is the first-filed court with jurisdiction over the dispute between these two parties. Seven's declaratory judgment action concerning the two Visto patents, although filed before Visto's motion for leave to amend its counterclaim, was instituted only after Visto approached Seven to meet and confer about the filing of the motion for leave to amend. As a result, this court will grant Visto's motion for leave to amend. This order is without prejudice to Seven's right to move to modify the scheduling order issued in this case or for separate trials of the issues raised by Visto's amended answer and counterclaims.

SIGNED this 17th day of August, 2006.

T. John Ward

T. JOHN WARD
UNITED STATES DISTRICT JUDGE

# EXHIBIT D

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

| | | |
|---|---|---|
| VISTO CORPORATION,<br><br>Plaintiff,<br><br>v.<br><br>SMARTNER INFORMATION SYSTEMS, LTD,<br><br>Defendant. | §<br>§<br>§<br>§<br>§<br>§<br>§<br>§<br>§ | Civil Action No. 2:05-CV-00091 (TJW) |

**ORDER**

Having considered Visto's Notice Withdrawing Visto's Motion For Leave to Amend Complaint Pursuant to Fed. Rule Civ. Proc. 15(A) [Dkt. No. 44], the Court hereby DENIES Visto's Motion for Leave to Amend Complaint Pursuant to Fed. Rule Civ. Proc. 15(A) as moot.

SIGNED this 31st day of January, 2007.

T. John Ward

T. JOHN WARD
UNITED STATES DISTRICT JUDGE